

[54] SYSTEM AND METHOD FOR TRANSMISSION OF DATA 5,311,570 5/1994 Grimes et al. 379/57

(List continued on next page.)

[75] Inventors: John M. Payne, Newport Beach; Tim Von Kaenel, Coto de Casa, both of Calif.; Jeffrey Wang, Horsham, Pa.; Jeffrey Odell, Newport Beach, Calif.; David Starr; Jason Katz, both of New York, N.Y.

[73] Assignee: Wireless Internet, Inc., New York, N.Y.

[21] Appl. No.: 08/788,613

[22] Filed: Jan. 24, 1997

Related U.S. Application Data

[60] Provisional application No. 60/010,651, Jan. 26, 1996, provisional application No. 60/014,341, Mar. 29, 1996, provisional application No. 60/014,735, Apr. 1, 1996, and provisional application No. 60/026,471, Sep. 23, 1996.

[51] Int. Cl.⁷ G06F 15/16; H04Q 1/30

[52] U.S. Cl. 709/219; 709/236; 340/311.1

[58] Field of Search 395/200.76, 200.49, 395/200.33; 709/203, 219, 236; 340/311.1

References Cited

U.S. PATENT DOCUMENTS

H1173	4/1993	Davis et al.	340/825.44
H1641	4/1997	Sharman	379/60
4,356,546	10/1982	Whiteside et al.	364/200
4,893,339	1/1990	Bright et al.	380/28
5,008,926	4/1991	Misholi	379/89
5,008,935	4/1991	Roberts	380/29
5,010,317	4/1991	Schwendeman et al.	340/311.1
5,043,721	8/1991	May	340/825.44
5,045,852	9/1991	Mitchell et al.	341/51
5,109,486	4/1992	Seymour	395/200
5,136,523	8/1992	Landers	395/54
5,140,419	8/1992	Galumbeck et al.	358/142
5,229,768	7/1993	Thomas	341/51
5,247,614	9/1993	Eagen et al.	395/200
5,253,294	10/1993	Maurer	380/21
5,281,967	1/1994	Jung	341/55
5,295,244	3/1994	Dev et al.	395/161

OTHER PUBLICATIONS

Ammons, B., "RBDS for Your Station!" <http://www.crlsystems.com/crl/tech/rbds.htm>, Circuit Research Labs, Inc., pp. 1-13, Sep. 1995.

Beam, M., "A Summary of Data Broadcasting Technologies and Potential Applications in Today's Market," http://www.dungeon.com/~start/rds_0030.html, Milestone Technologies, Inc., pp. 1-8, Dec. 1994.

Hemingway, D., "Radio Data System," <http://www.dungeon.com/~start/rdsdh.html>, pp. 1-9, Dec. 1995.

Motorola, Inc. Short Form Address with Message Protocol, Jun. 24, 1995. http://www.mot.com/MIMS/PPG/products/2way/reflex/tam_isv/sfp_s2.html, see entire document.

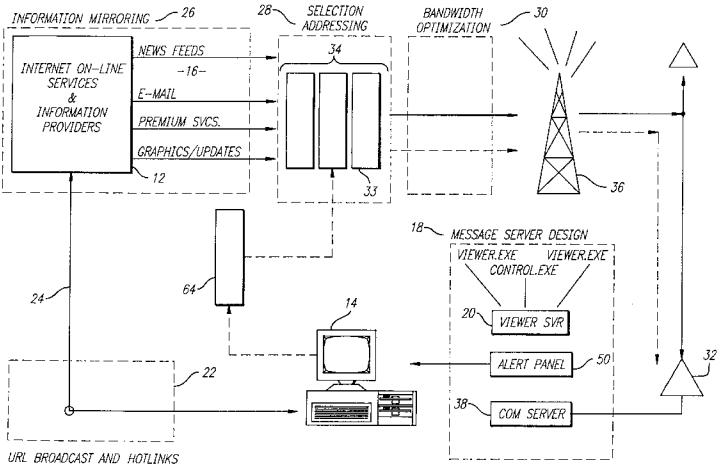
Motorola, Inc. Short Form Address with Message Protocol, Apr. 21, 1996. http://www.mot.com/MIMS/MSPG/CTSD/products/short_form/sfp_s2.html, see entire document.

Primary Examiner—Zarni Maung
Assistant Examiner—Andrew Caldwell
Attorney, Agent, or Firm—Akin, Gump, Strauss, Hauer & Feld, L.L.P.

[57] ABSTRACT

A system and method for data communication connecting on-line networks with on-line and off-line computers. The present system provides for broadcast of up to the minute notification centric data thereby providing an instant call to action for users who are provided with the ability to instantaneously retrieve further detailed information. Information sources transmit data to a central broadcast server, which preprocesses the data for wireless broadcast. The notification centric portions of data are wirelessly broadcast to wireless receiving devices that are attached to computing devices. Upon receipt of the data at the computing device, the user is notified through different multimedia alerts that there is an incoming message. Wirelessly broadcasted URL's, associated with the data, are embedded in data packets and provide an automated wired or wireless connection back to the information source for obtaining detailed data.

77 Claims, 30 Drawing Sheets



6,021,433

Page 2

U.S. PATENT DOCUMENTS

5,315,635	5/1994	Kane et al.	379/57	5,544,354	8/1996	May et al.	395/600
5,325,505	6/1994	Hoffecker et al.	395/425	5,548,814	8/1996	Lorang et al.	455/38.1
5,341,425	8/1994	Wasilewski et al.	380/20	5,555,446	9/1996	Jasinski	455/54.2
5,347,632	9/1994	Filepp et al.	395/200	5,572,643	11/1996	Judson	395/793
5,367,670	11/1994	Ward et al.	395/575	5,581,576	12/1996	Lanzetta et al.	375/216
5,371,499	12/1994	Graybill et al.	341/51	5,592,512	1/1997	Spiess	375/240
5,377,354	12/1994	Scannell et al.	395/650	5,596,318	1/1997	Mitchell	340/825.44
5,392,451	2/1995	Schwendeman et al.	455/13.1	5,596,694	1/1997	Capps	395/152
5,394,522	2/1995	Sanchez-Frank et al.	395/159	5,602,596	2/1997	Claussen et al.	348/564
5,394,537	2/1995	Schwendeman	379/57	5,606,569	2/1997	MacDonald et al.	371/37.7
5,396,560	3/1995	Kane	455/12.1	5,614,899	3/1997	Tokuda et al.	341/51
5,398,021	3/1995	Moore	340/825.27	5,617,541	4/1997	Albanese et al.	395/200.13
5,406,278	4/1995	Graybill et al.	341/51	5,617,565	4/1997	Augenbraun et al.	395/604
5,406,279	4/1995	Anderson et al.	341/51	5,623,589	4/1997	Needham et al.	395/160
5,423,086	6/1995	Cannon et al.	455/186	5,623,613	4/1997	Rowe et al.	395/353
5,442,652	8/1995	Jacobson	375/200	5,627,978	5/1997	Altom et al.	395/330
5,446,885	8/1995	Moore et al.	395/600	5,634,062	5/1997	Shimizu et al.	395/762
5,452,356	9/1995	Albert	380/9	5,701,302	12/1997	Geiger	123/48 B
5,459,458	10/1995	Richardson et al.	340/825.52	5,732,338	3/1998	Schwob	455/158.5
5,481,255	1/1996	Albert et al.	340/825.55	5,790,958	8/1998	McCoy et al.	455/557
5,493,692	2/1996	Theimer et al.	455/26.1	5,796,394	8/1998	Wicks et al.	345/331
5,504,476	4/1996	Marrs et al.	340/825.44	5,809,415	9/1998	Rossman	455/422
5,510,778	4/1996	Krieter et al.	340/825.44	5,838,252	11/1998	Kikinis	340/825.44
5,510,808	4/1996	Cina, Jr. et al.	345/123	5,872,921	2/1999	Zahariev et al.	709/203
5,526,363	6/1996	Weiss et al.	370/118	5,892,432	4/1999	Skoog	340/313
5,530,645	6/1996	Chu	364/419.13	5,903,262	3/1999	Ichihashi et al.	345/327
5,532,694	7/1996	Mayers et al.	341/67	5,905,865	5/1999	Palmer et al.	709/217
5,537,586	7/1996	Amram et al.	395/600	5,907,793	5/1999	Reams	455/3.1
5,543,781	8/1996	Ganucheau, Jr. et al.	340/825.44	5,909,651	6/1999	Chander et al.	455/466
5,544,320	8/1996	Konrad	395/200.09	5,954,793	9/1999	Stutman et al.	709/204

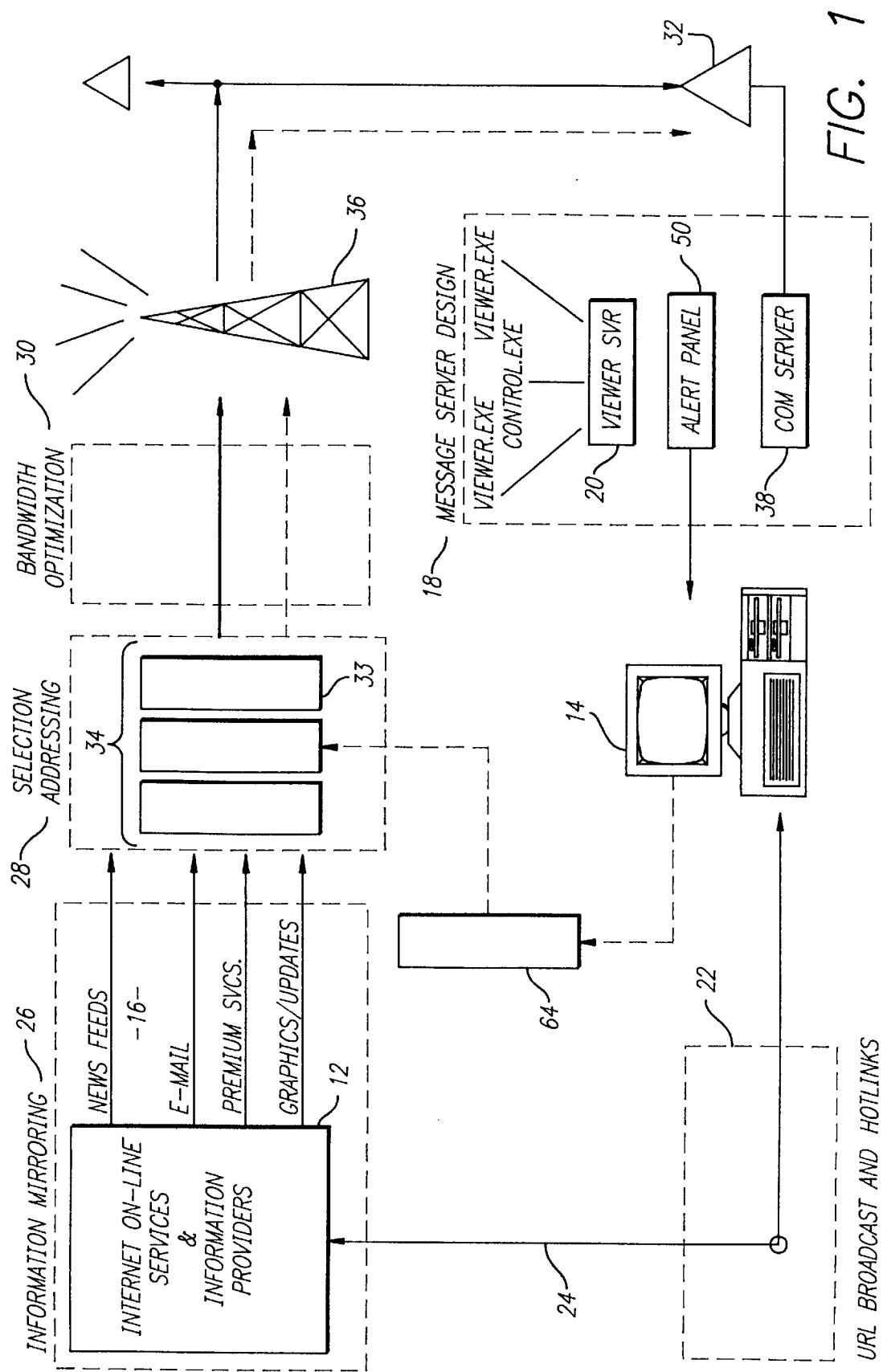
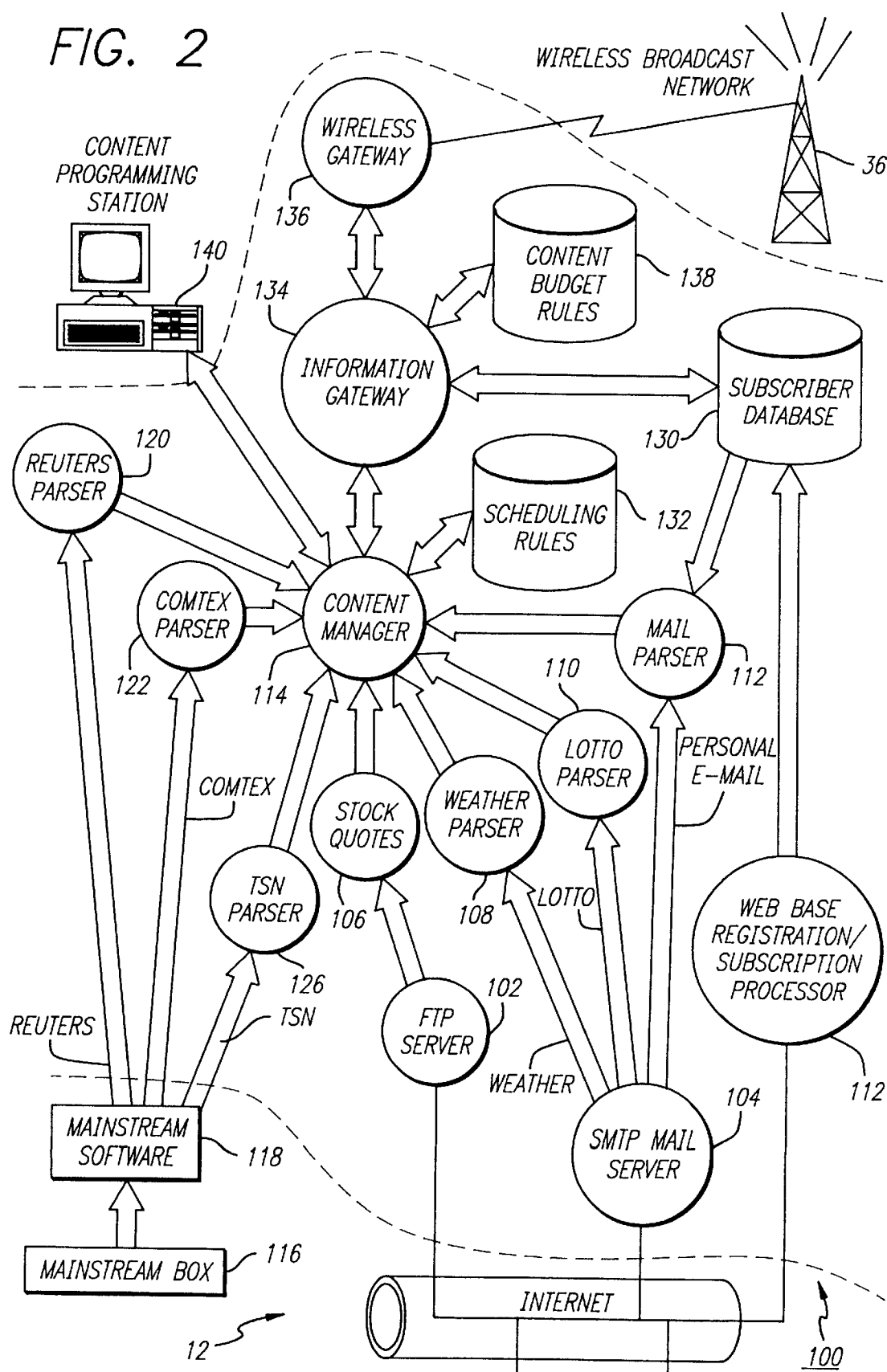


FIG. 2



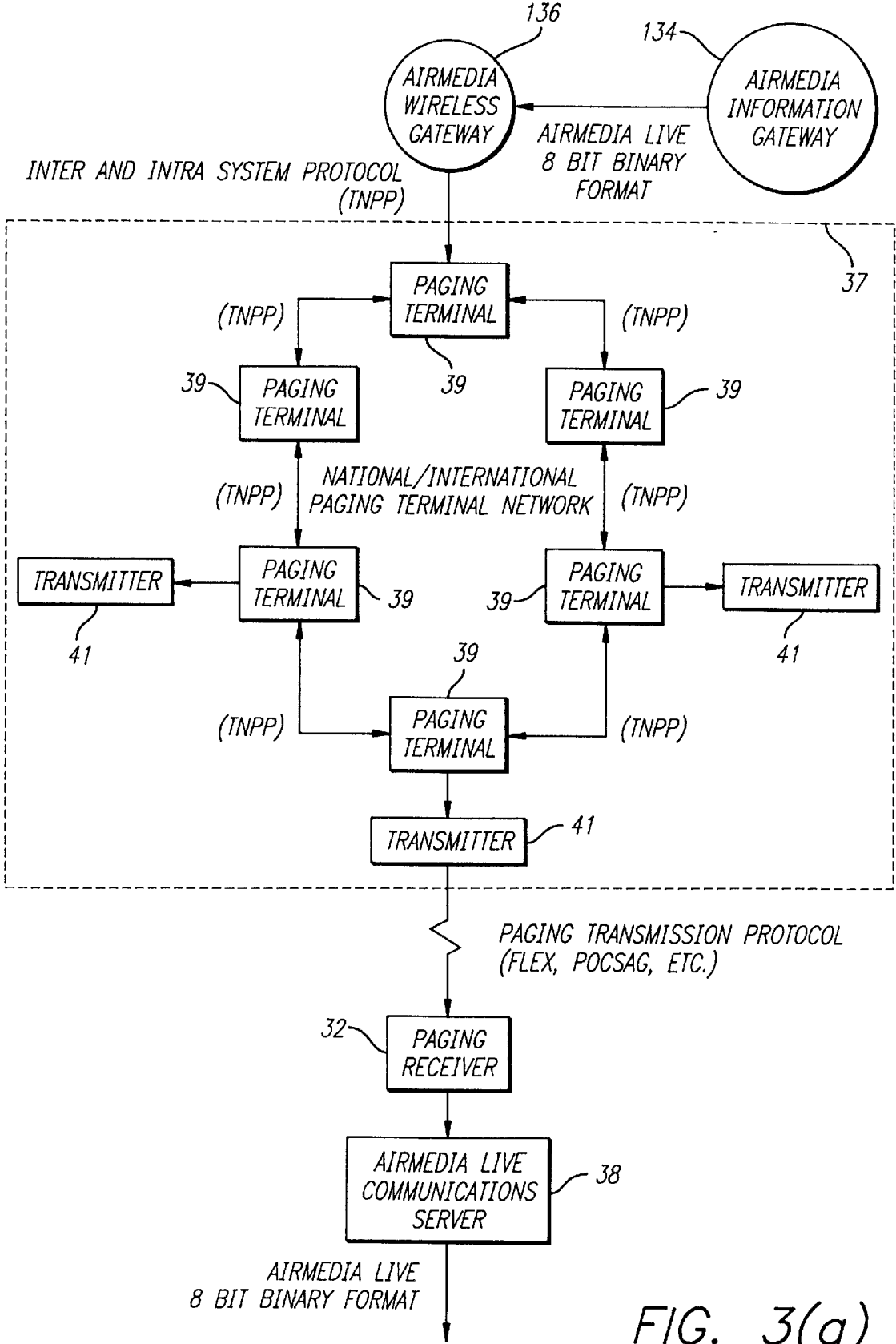


FIG. 3(a)

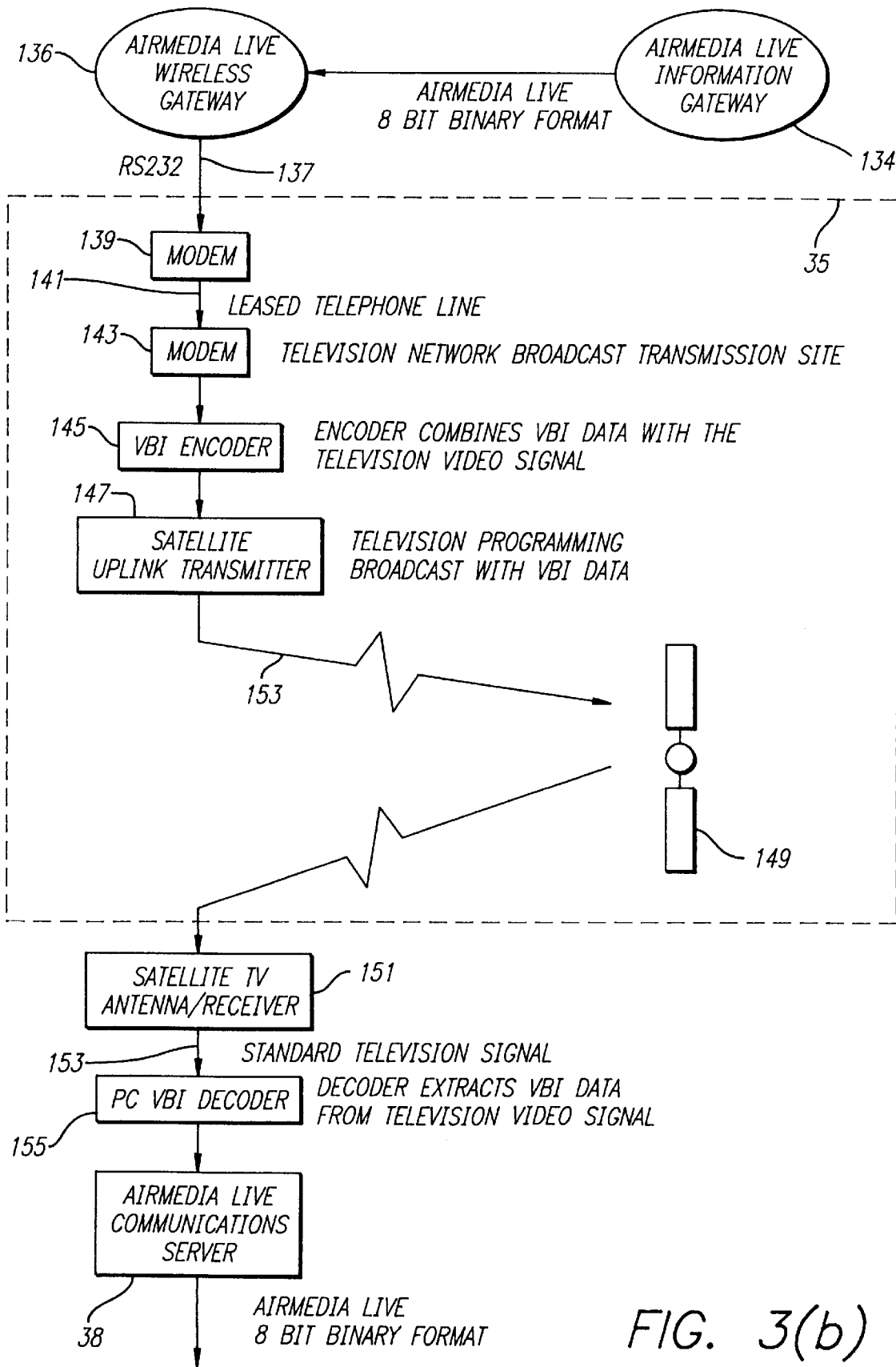


FIG. 3(b)

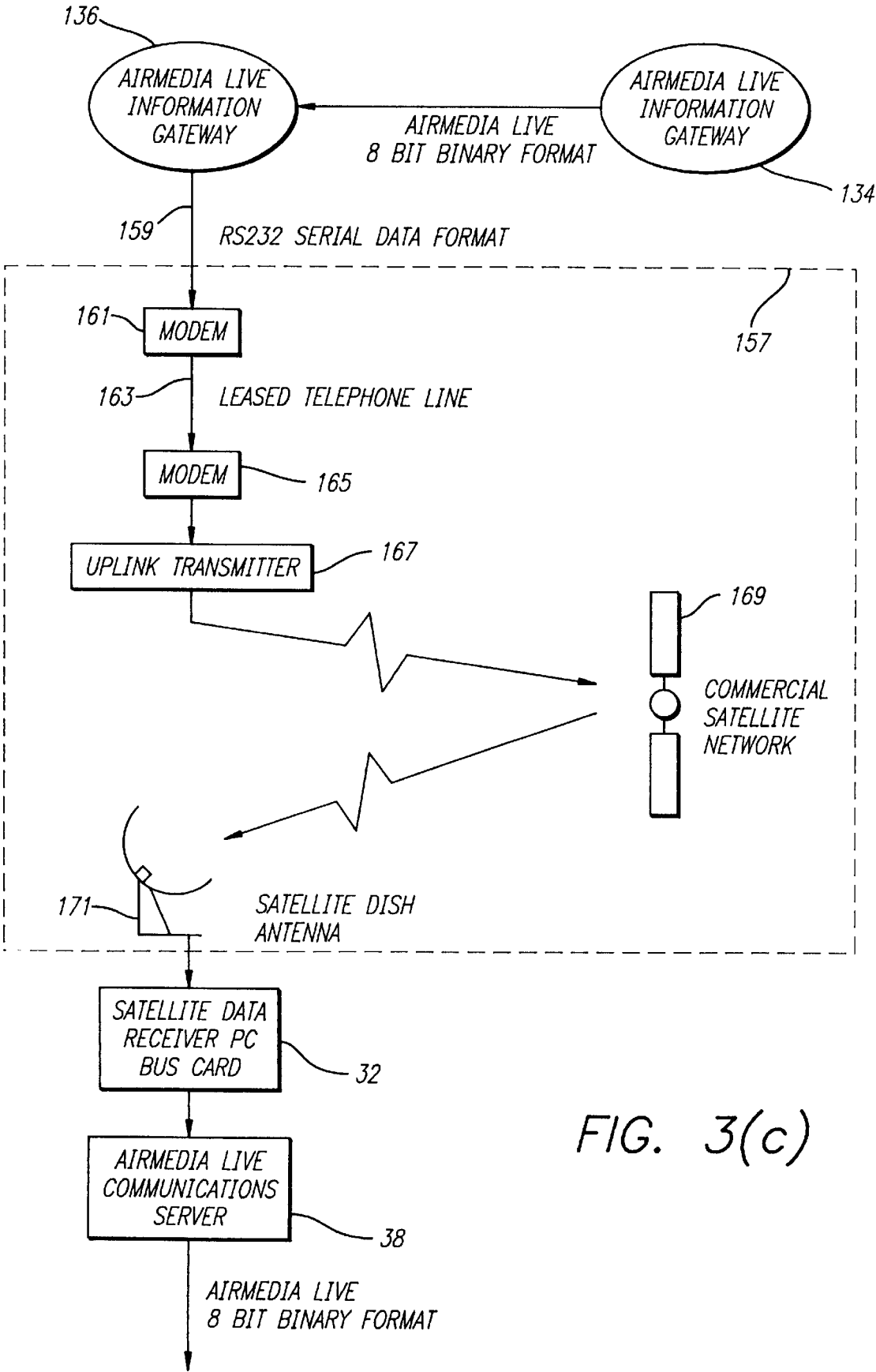


FIG. 3(c)

U.S. Patent

Feb. 1, 2000

Sheet 6 of 30

6,021,433

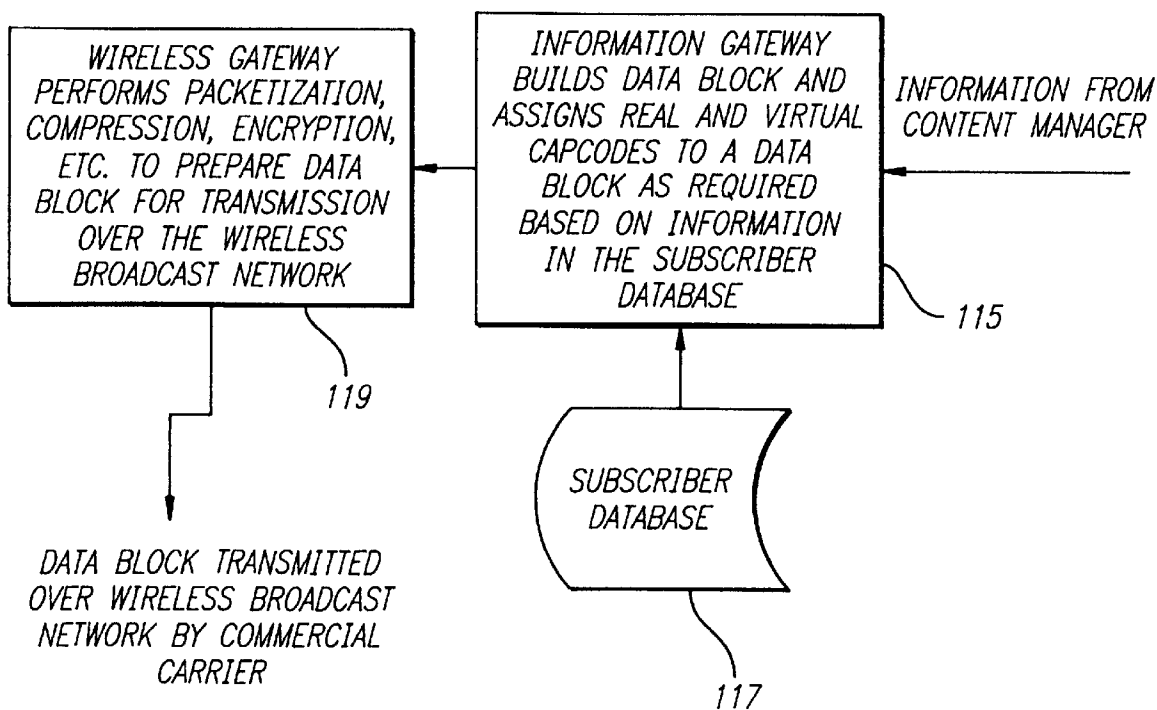


FIG. 4

FIG. 5-1

7/30

Item	Size	Description
Header:		
CRC	2 bytes	Standard Cyclical Redundancy Code to verify data block integrity.
Header Type	1 bit	If bit clear, then this is a message header. If bit set, then this is the data block header.
Custom Header Flag	1 bit	If bit clear, no custom header. If bit set, then a custom header is included in the data block.
Version Number	4 bits	Protocol version used.
Private Data Block Flag	1 bit	If bit clear, then this data block will be passed on to the Alert Panel for processing and display. If bit set, then this is a private data block to be processed internally by the Communications Server.
Virtual Capcode Flag	1 bit	If bit clear, then this data block is not targeted for a specific virtual capcode and no virtual capcode is included in the data block. If bit set, then this data block contains a virtual capcode.
Data Block Type	1 byte	The value of this byte specifies the type of data contained in the data block. If Private Data Block Flag is clear: 1 = plain text, 2 = AirMedia Live data feed format. If Private Data Block Flag is set: 1 = Capcode reprogramming message, 2 = Binary file transfer.
Data Block Version	4 bits	The version number of this data block's format.

FIG. 5-2

Use Compression Flag	1 bit	If bit clear, then this data block is not compressed. If bit set, then compression is used and the compression type is specified in the Compression ID item.
Use Encryption Flag	1 bit	If bit clear, then this data block is not encrypted. If bit set, then this data block is encrypted.
Spare	2 bits	Reserved for future use.
Compression ID (optional)	1 byte	Included only if Use Compression Flag is set. Indicates the type of compression used.
Virtual Capcode (optional)	1 byte	Included only if Virtual Capcode flag is set. Contains the virtual capcode to which this data block is targeted.
Size of Custom Header (optional)	1 byte	Included only if Custom Header Flag is set. Contains the size in bytes of the custom header.
Custom Header (optional)	variable	Reserved for future enhancements to data block protocol. Size determined from previous item.
Contents:		
Data Block Contents	variable	Information notification data from the information source to be processed by AirMedia Live software.

FIG. 6

Item	Size	Description
Header:		
Alert Length	1 byte	The size of the alert data in bytes.
Alert Type	1 byte	The value of this item defines the alert type (e.g. new e-mail arrival alert). Up to 256 predefined alert types are allowed.
Contents:		
Alert Data	variable	Personal alert notification data. Size of data is determined by the Alert Length item.

FIG. 9

Item	Size	Description
Header:		
Packet Type	4 bits	The value of this item indicates the packet type: 0 = Standard AirMedia Live Packet; 1 = Single Packet Data Block; if the left most bit (high bit) is set, then this is a Binary Alert Packet.
Data Block ID	12 bits	The ID of the data block contained in this packet.
Contents:		
Packet Contents	variable	The header and contents of the data block contained in this packet.

FIG. 7

Item	Size	Description
Header:		
CRC	2 bytes	Standard Cyclical Redundancy Code to verify message integrity.
Header Type	1 bit	If bit clear, then this is a message header. If bit set, then this is the data block header.
Custom Header Flag	1 bit	If bit clear, no custom header. If bit set, then a custom header is included in the message.
Data Block ID	14 bits	ID of the data block to which this message belongs.
Message Number	1 byte	The position of this message in the data block (i.e. message sequence number).
Total Messages	1 byte	Total number of messages in the data block.
Size of Custom Header (optional)	1 byte	Included only if Custom Header Flag is set. Contains the size in bytes of the custom header.
Custom Header (optional)	variable	Reserved for future enhancements to message protocol. Size determined from previous item.
Contents:		
Message Contents	variable	The data portion of the message.

FIG. 8

Item	Size	Description
Header:		
Packet Type	4 bits	The value of this item indicates the packet type: 0 = Standard AirMedia Live Packet; 1 = Single Packet Data Block; if the left most bit (high bit) is set, then this is a Binary Alert Packet.
Total Packets Flag	1 bit	If bit is clear, then the Total Data Packets and Total Error Correction Packets items are not present. If bit is set, then the Total Data Packets and Total Error Correction Packets items are present.
Message ID	11 bits	The number of the message to which this packet belongs.
Packet Number	1 byte	The position of this packet in the message (packet sequence number).
Total Data Packets	1 byte	Total number of data packets in the message (does not include error correction packets).
Total Error Correction Packets	1 byte	Total number of Reed-Solomon forward error correction packets in the message.
Contents:		
Packet Contents	variable	The data portion of the packet.

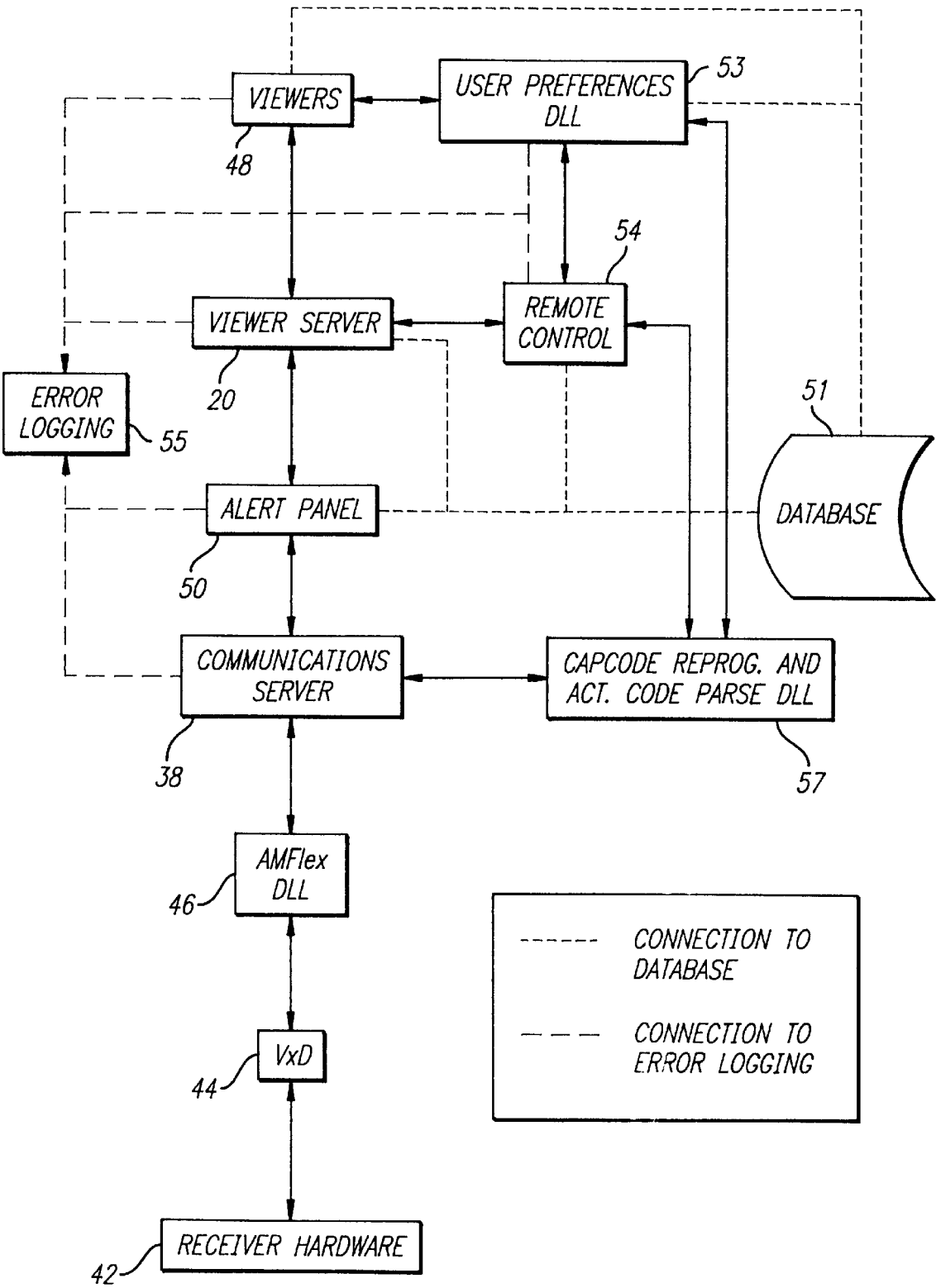


FIG. 10

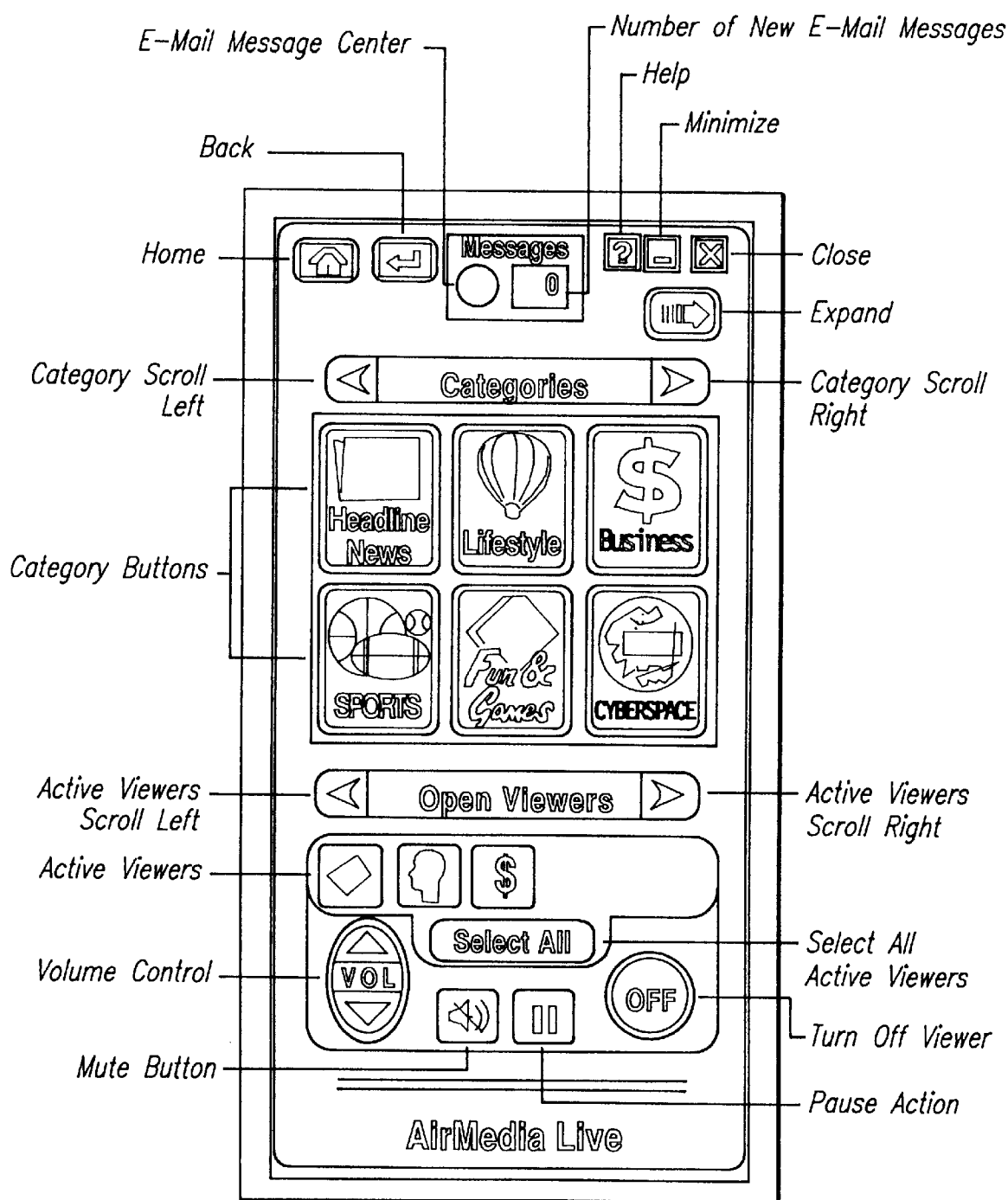


FIG. 11

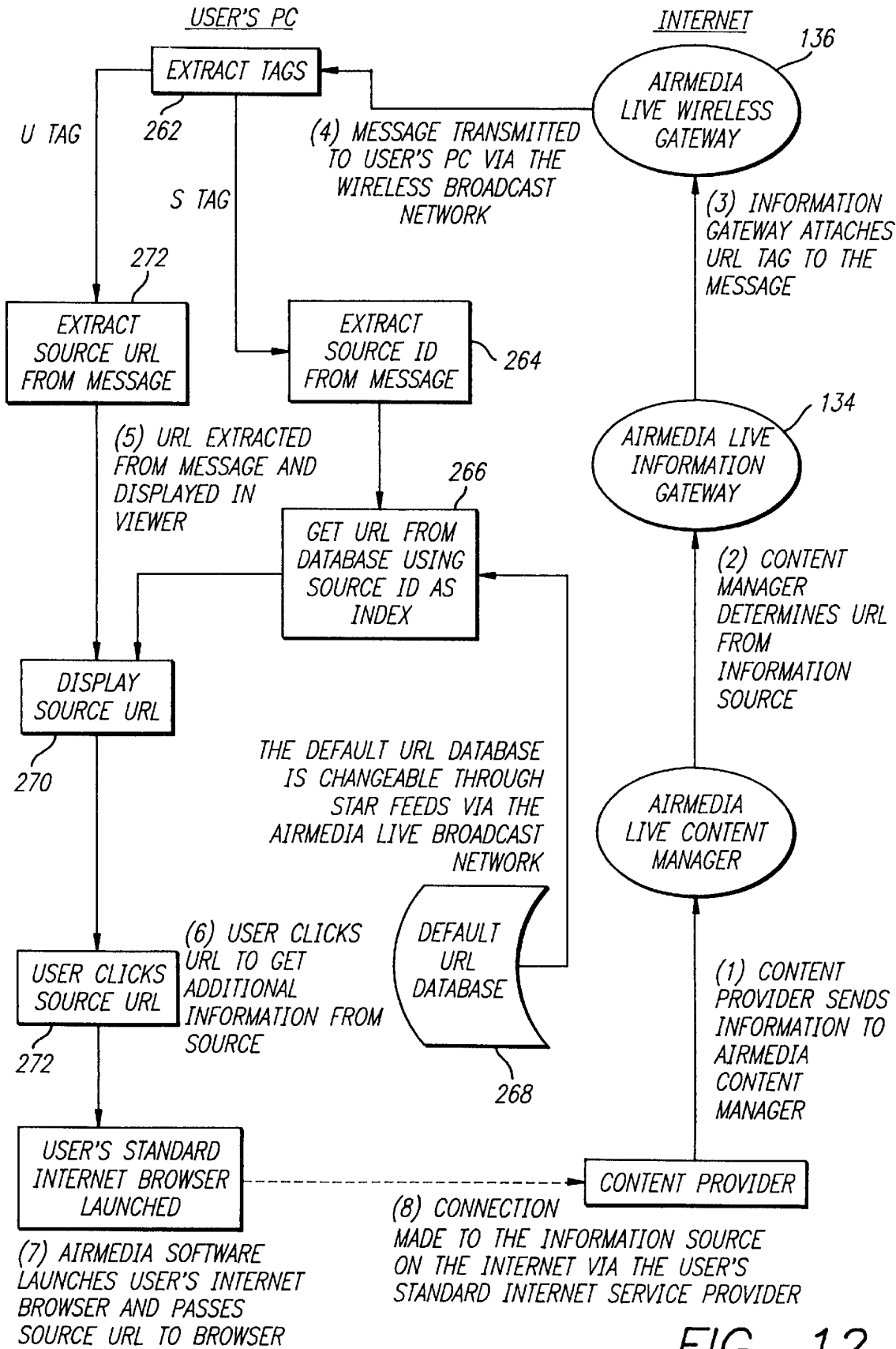


FIG. 12

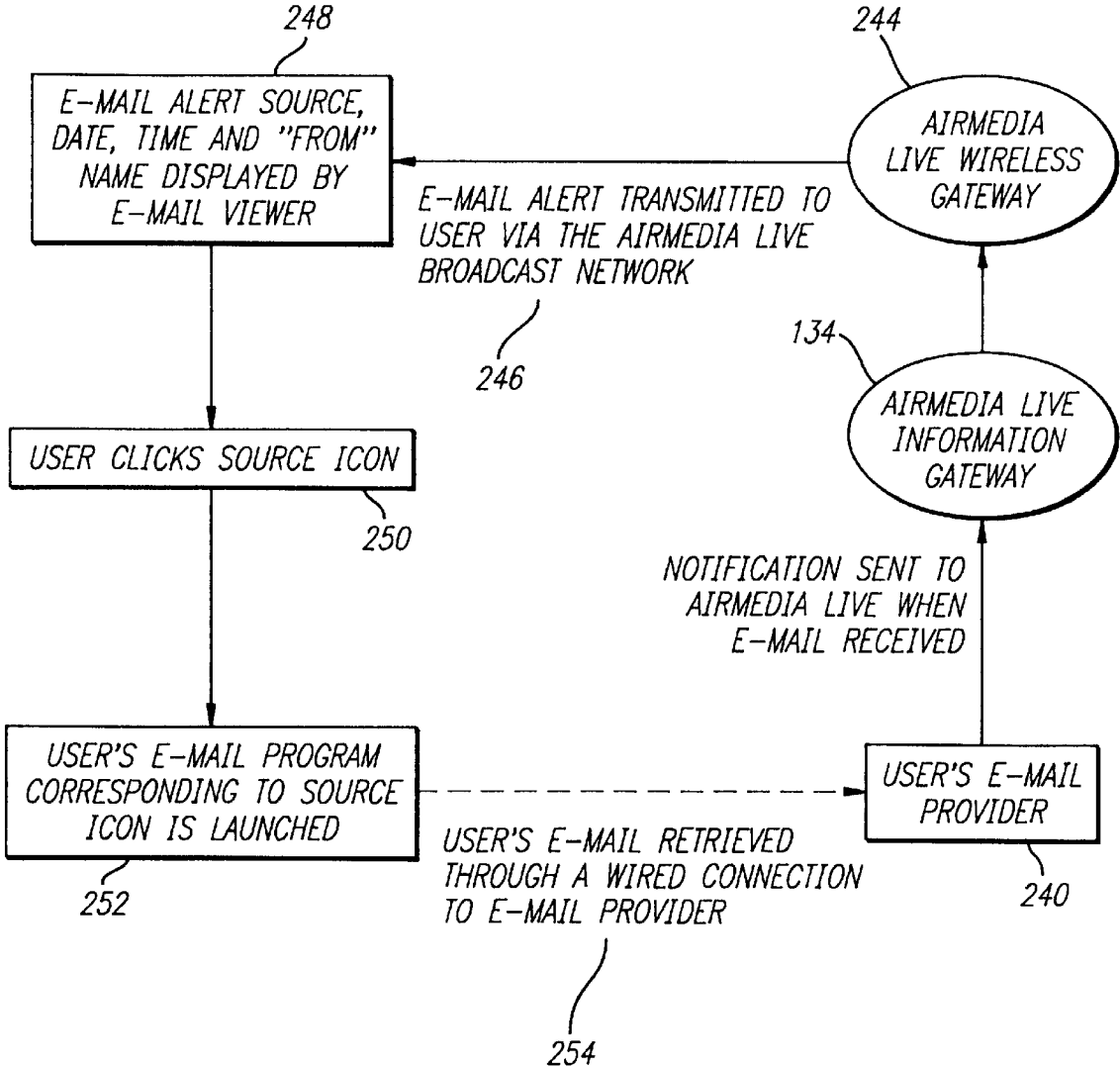
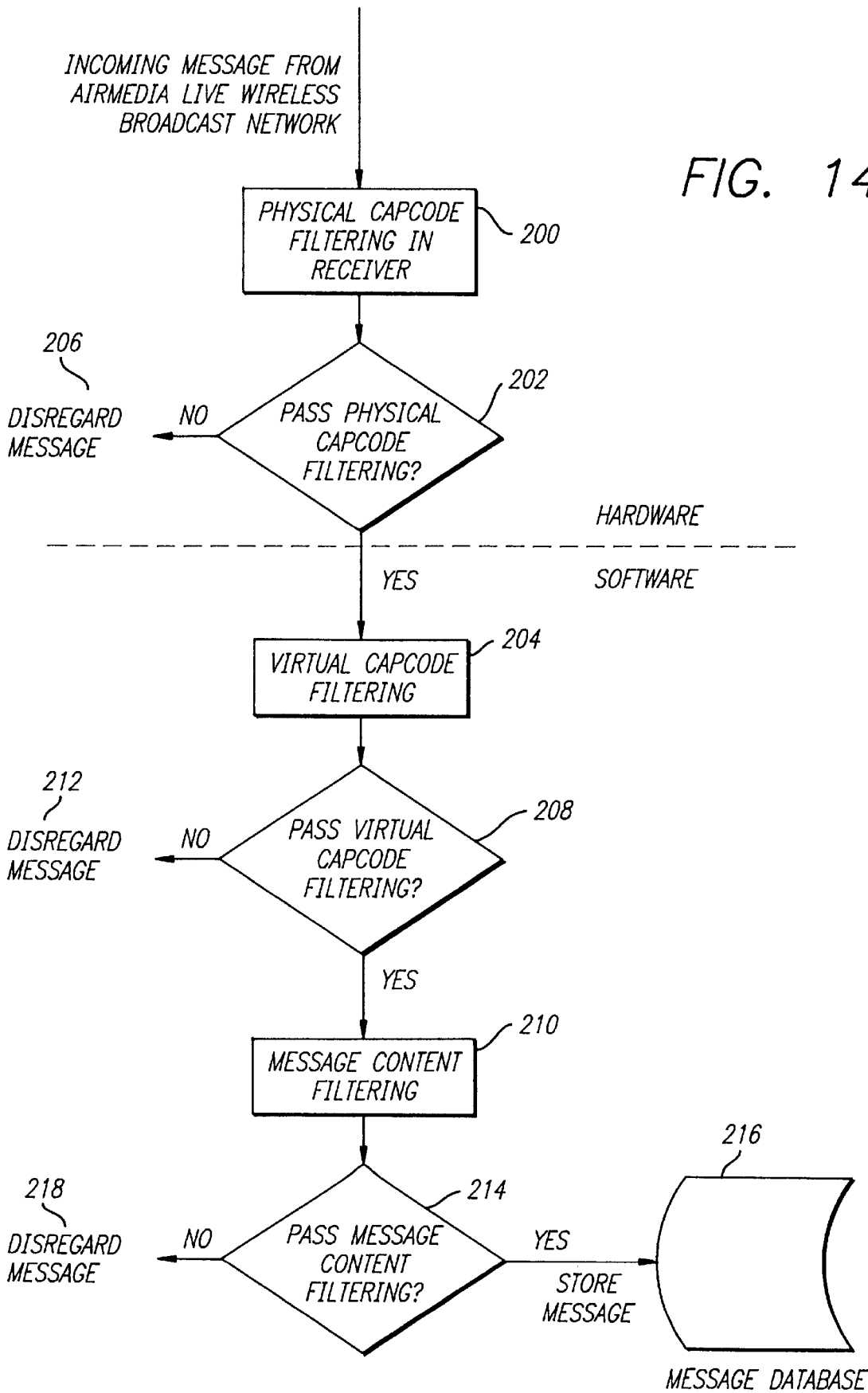


FIG. 13



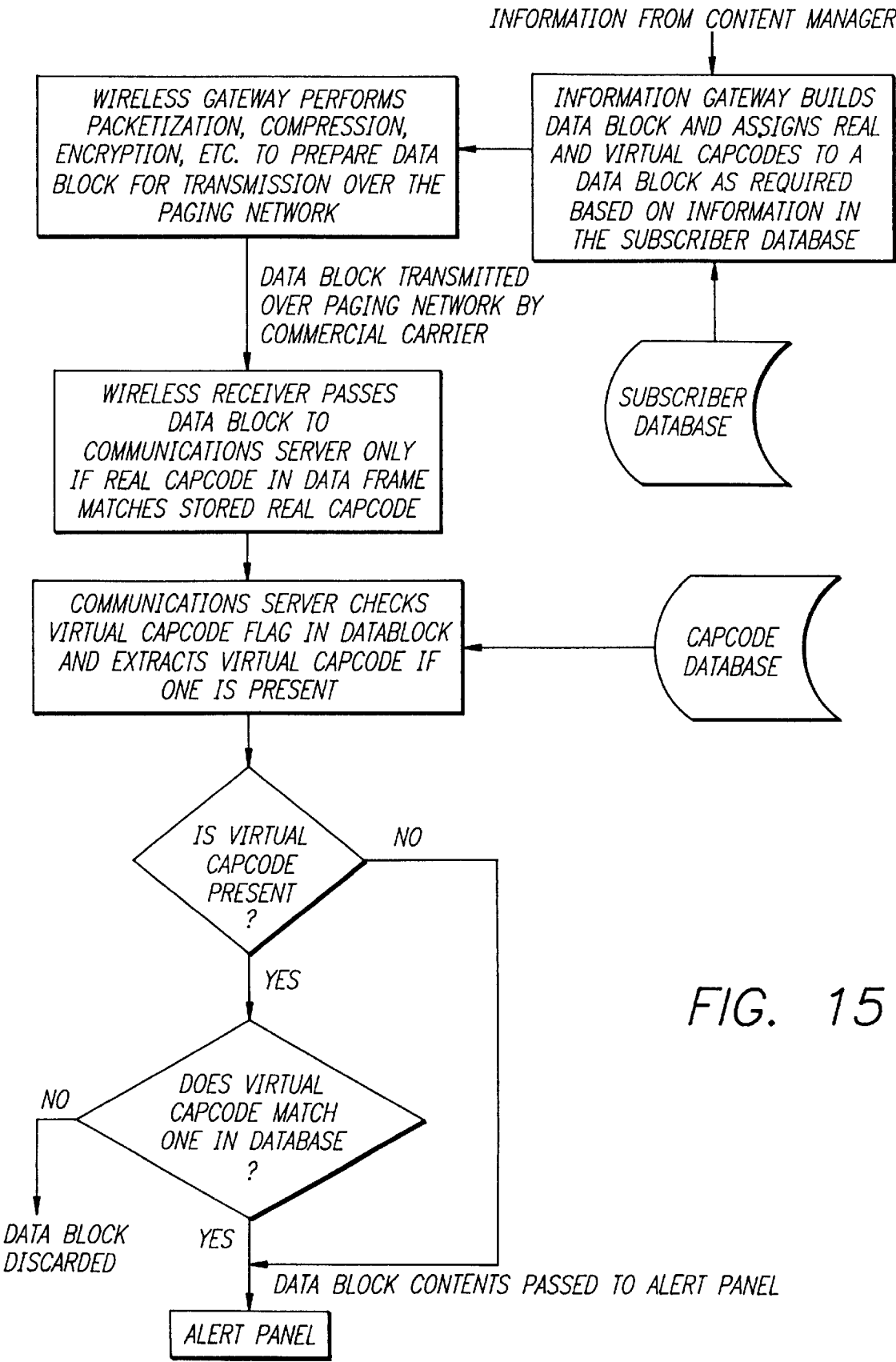


FIG. 15

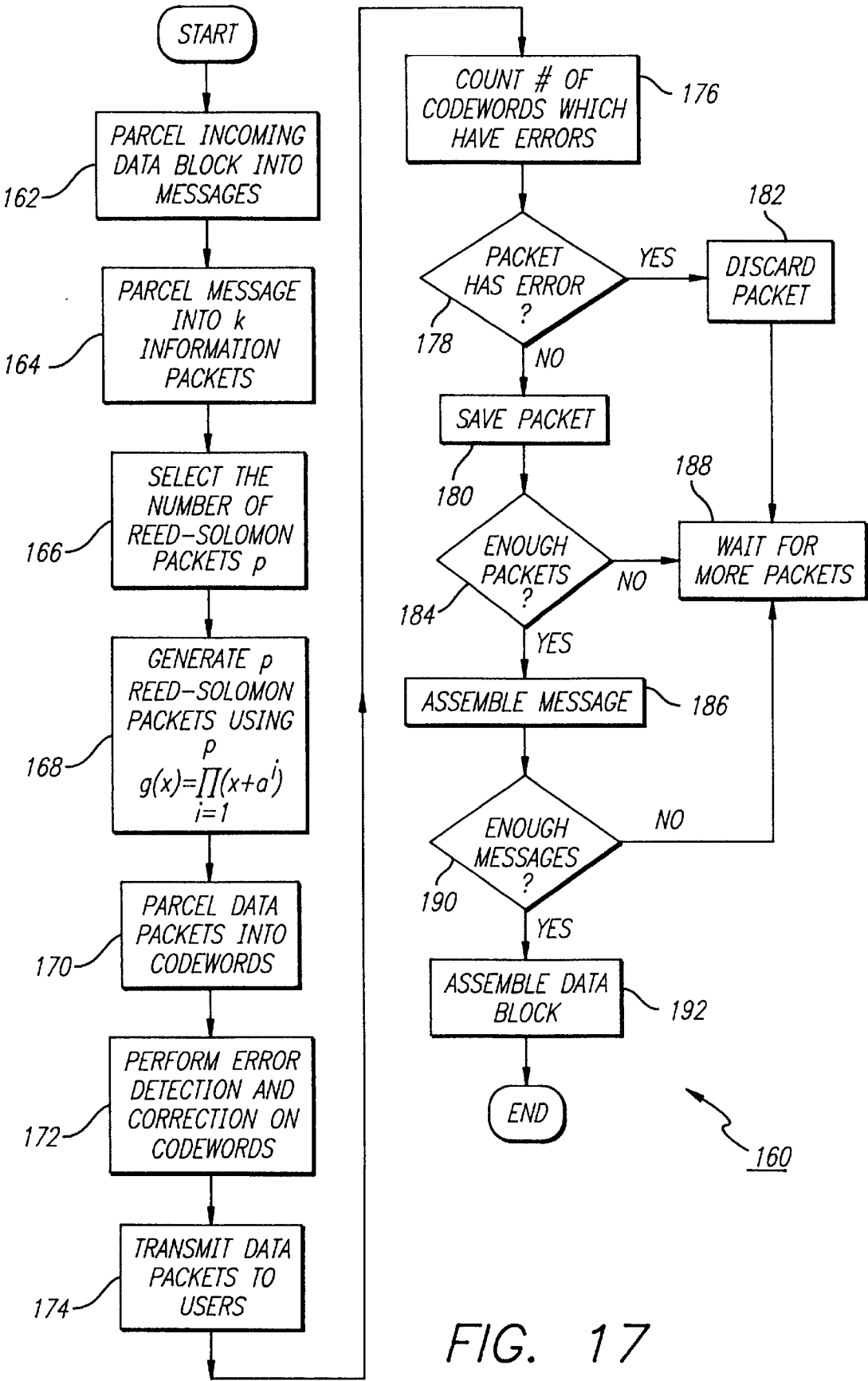
FIG. 16

Whole Packets	Packet Header	Packet	column 1	column 2	column 3	column s
packet 1:	header 1	x	x	x	x	x
packet 2:	header 2	x	x	x	x	x
.
.
.
packet p:	header p	x	x	x	x	x
packet p + 1	header p + 1	x	x	x	x	x
packet p + 2	header p + 2	x	x	x	x	x
packet p + 3	header p + 3	x	x	x	x	x
.
.
.
packet p + x	header p + x	x	x	x	x	x

Reed Solomon
Parity- Check
Packets

Information
Packets

150



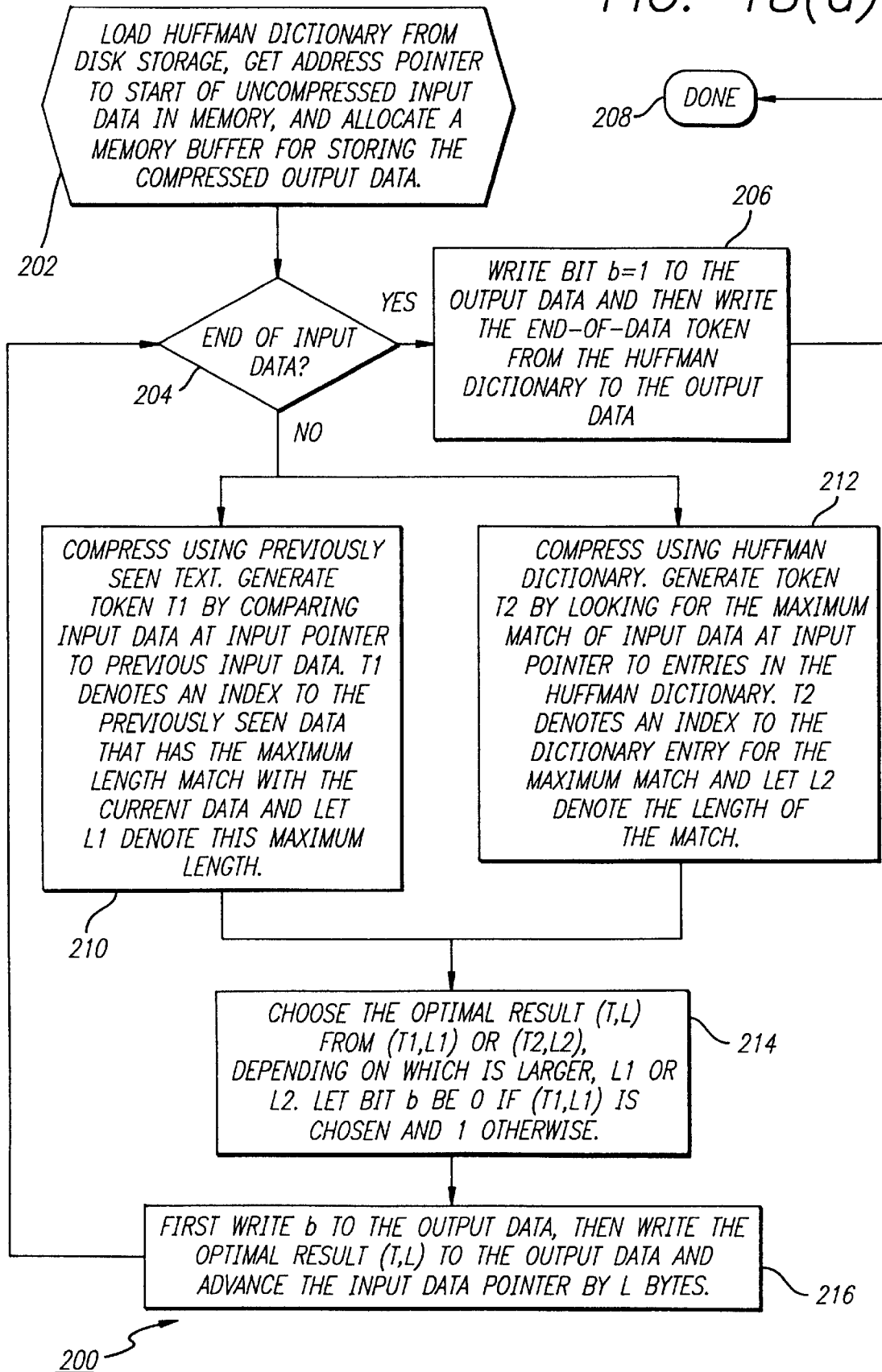
U.S. Patent

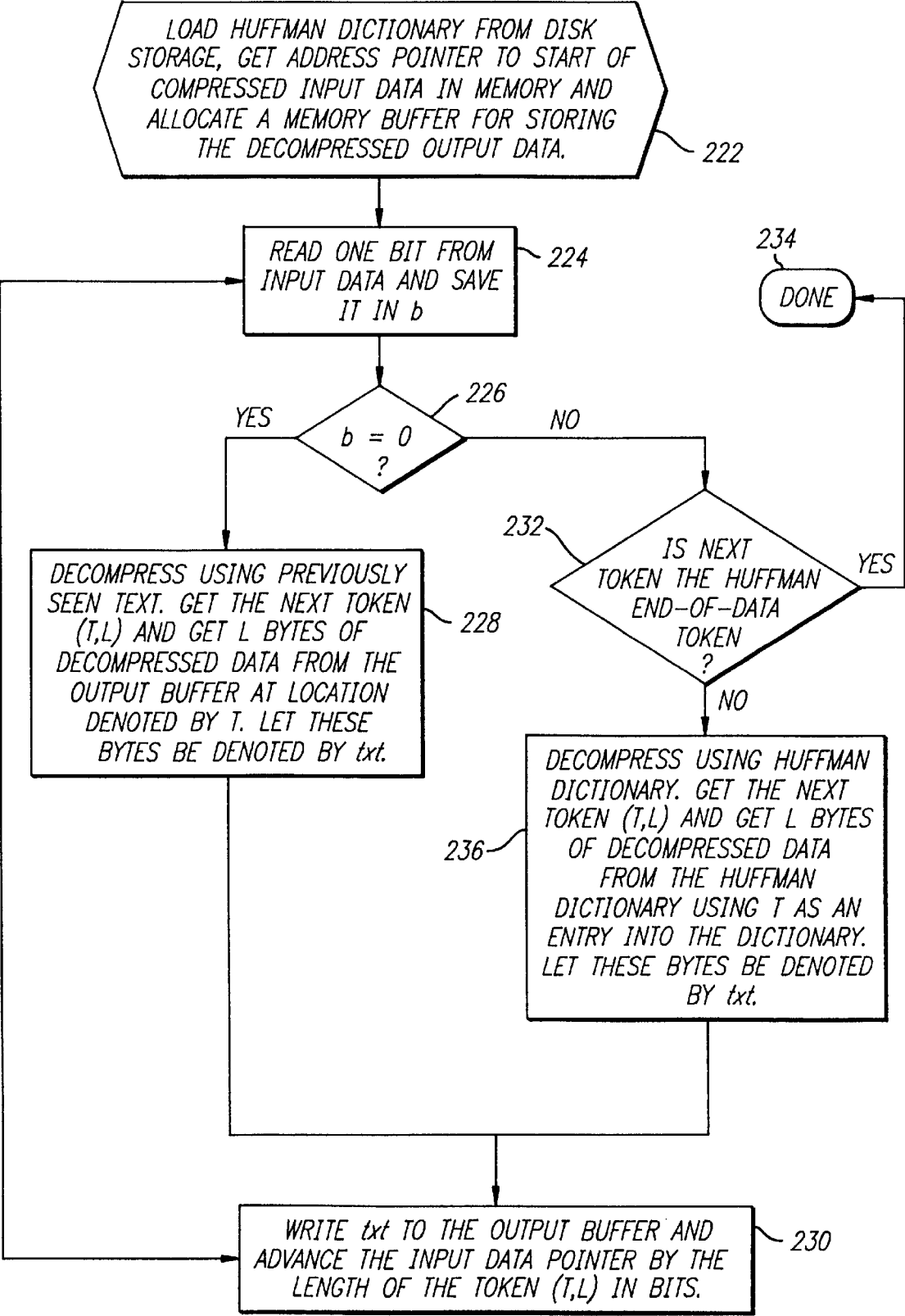
Feb. 1, 2000

Sheet 20 of 30

6,021,433

FIG. 18(a)





220

FIG. 18(b)

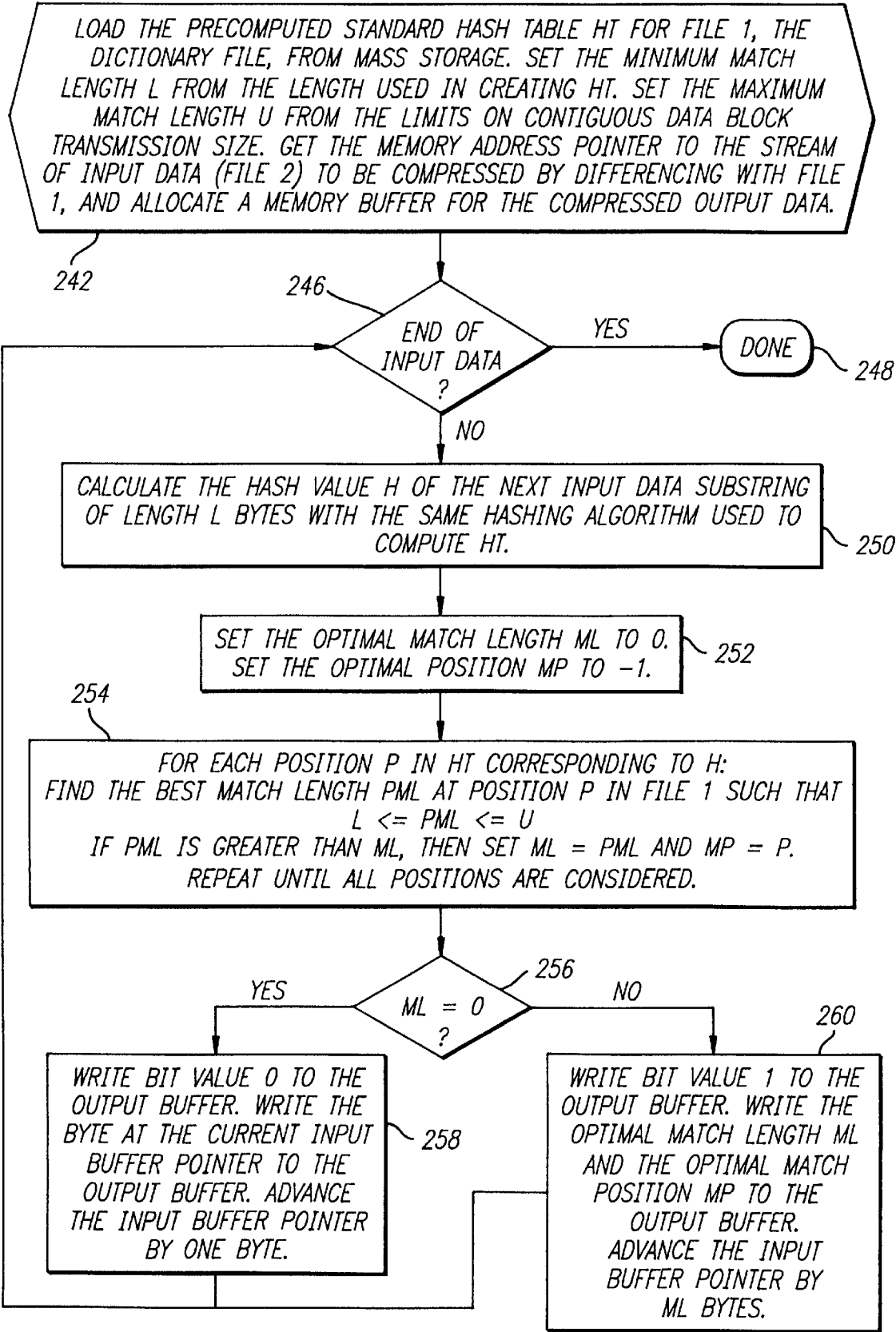


FIG. 19(a)

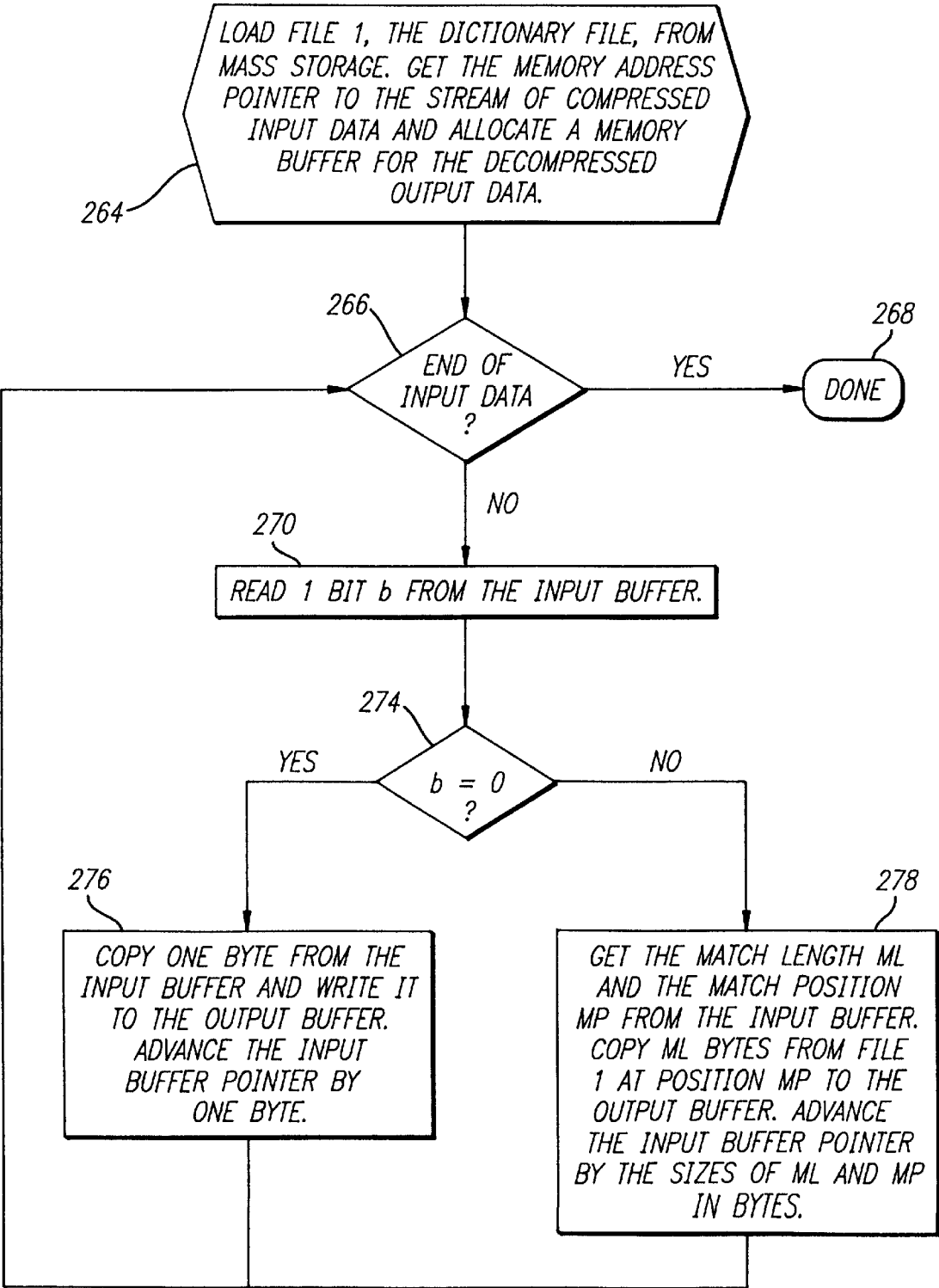


FIG. 19(b)

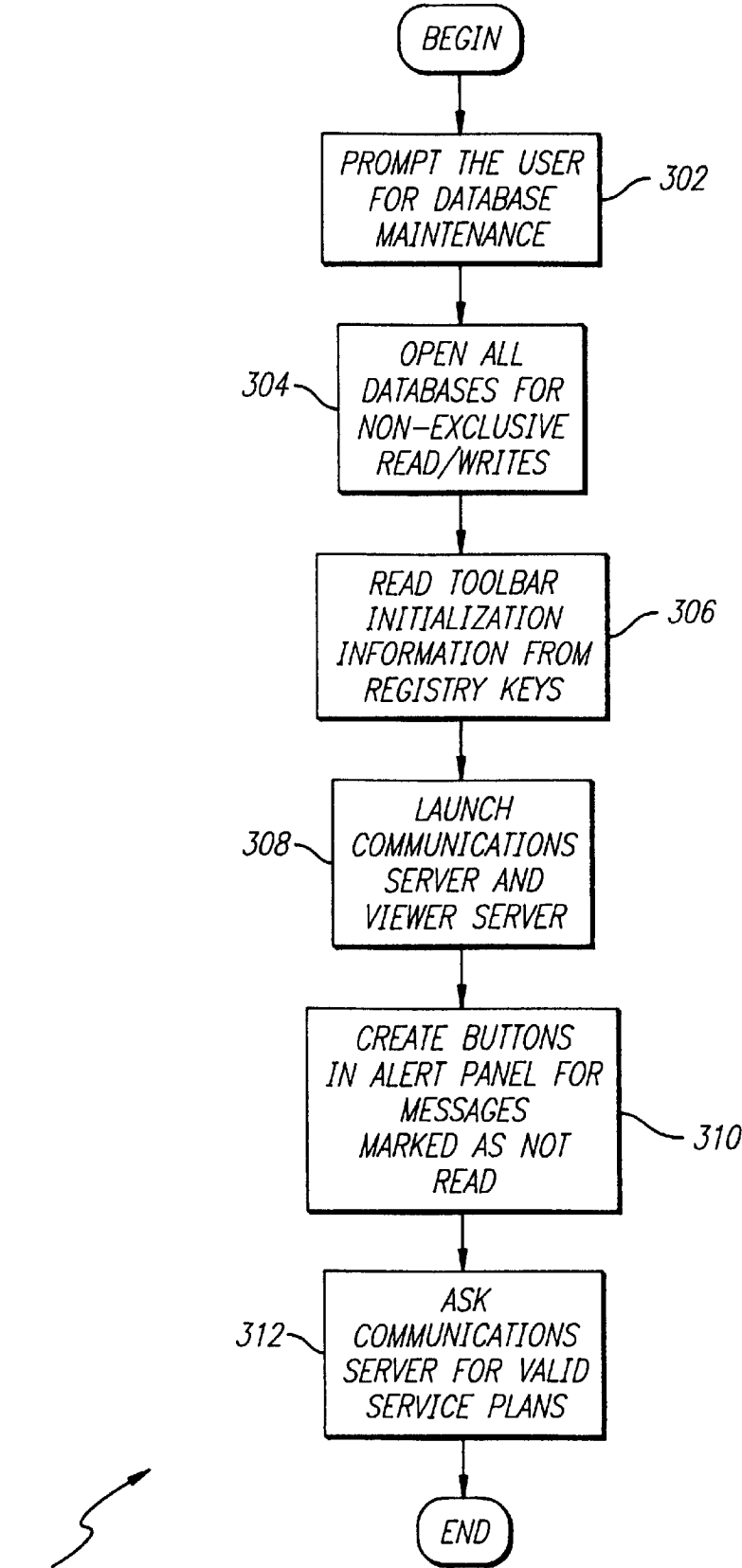


FIG. 21

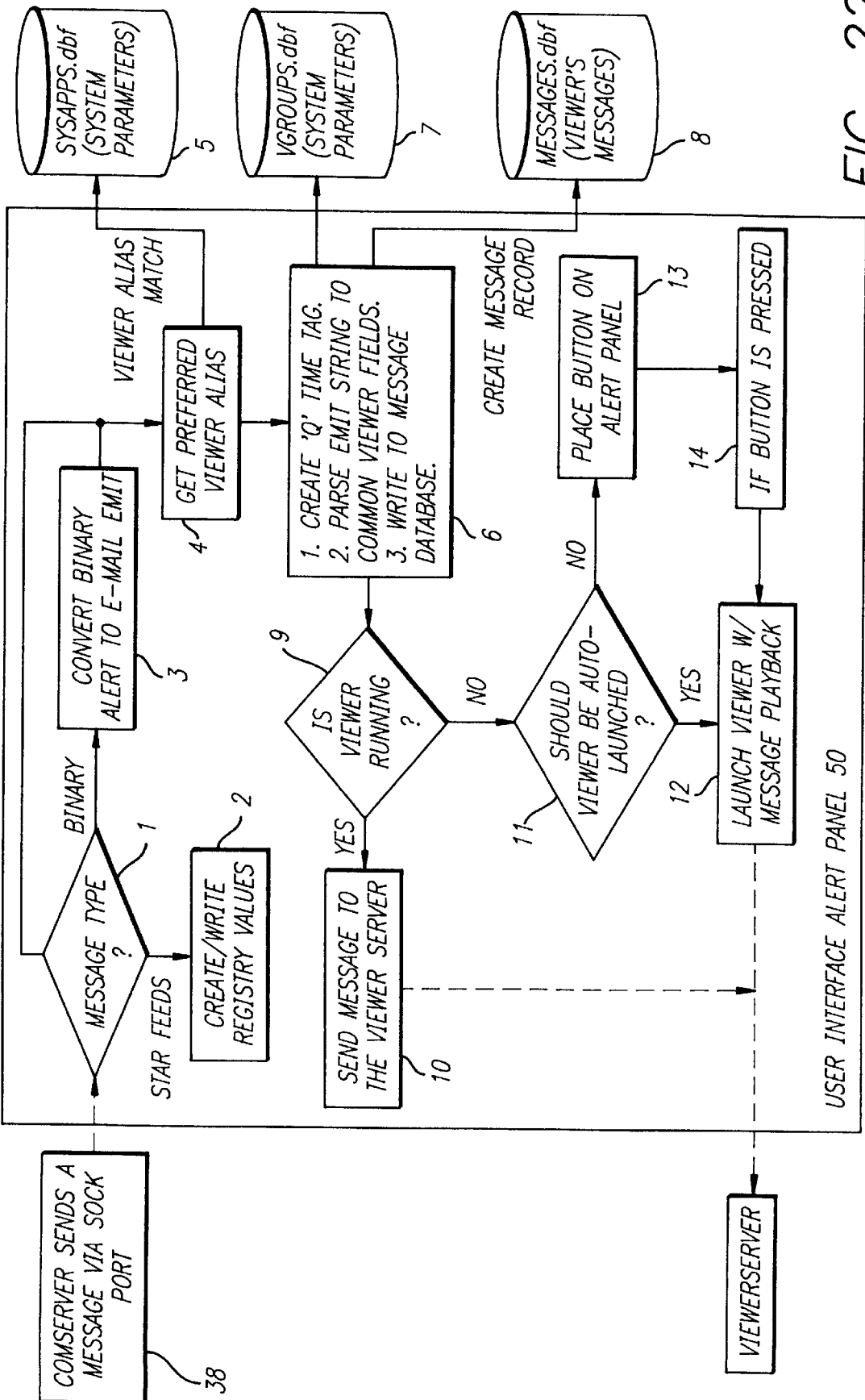


FIG. 22

FIG. 23

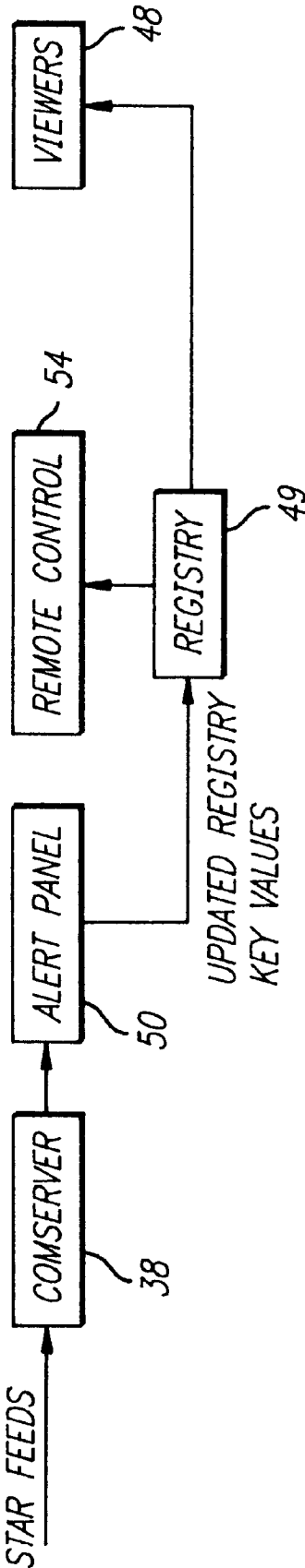


FIG. 24(a)

FIG. 24(b)

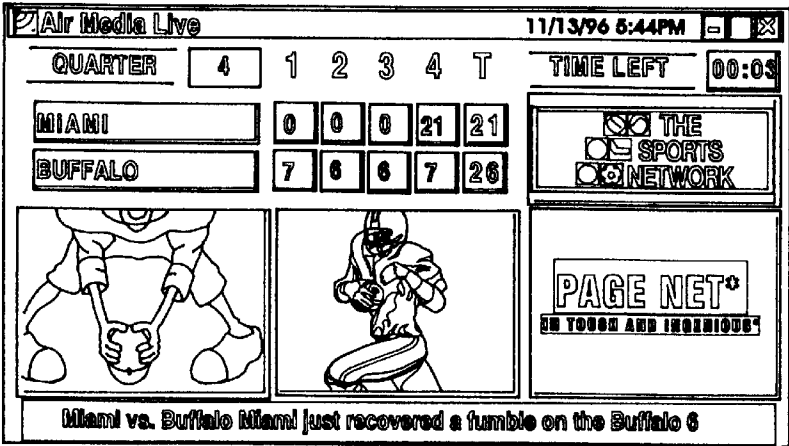


FIG. 24(c)

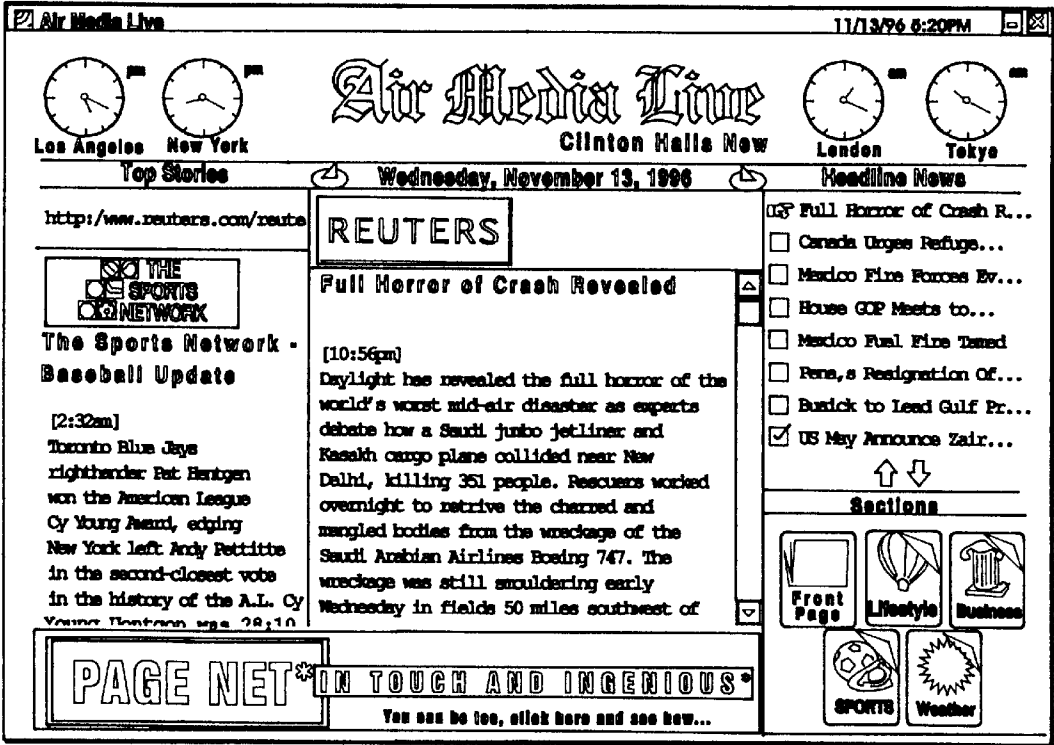
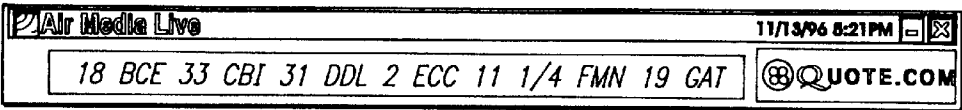


FIG. 24(d)



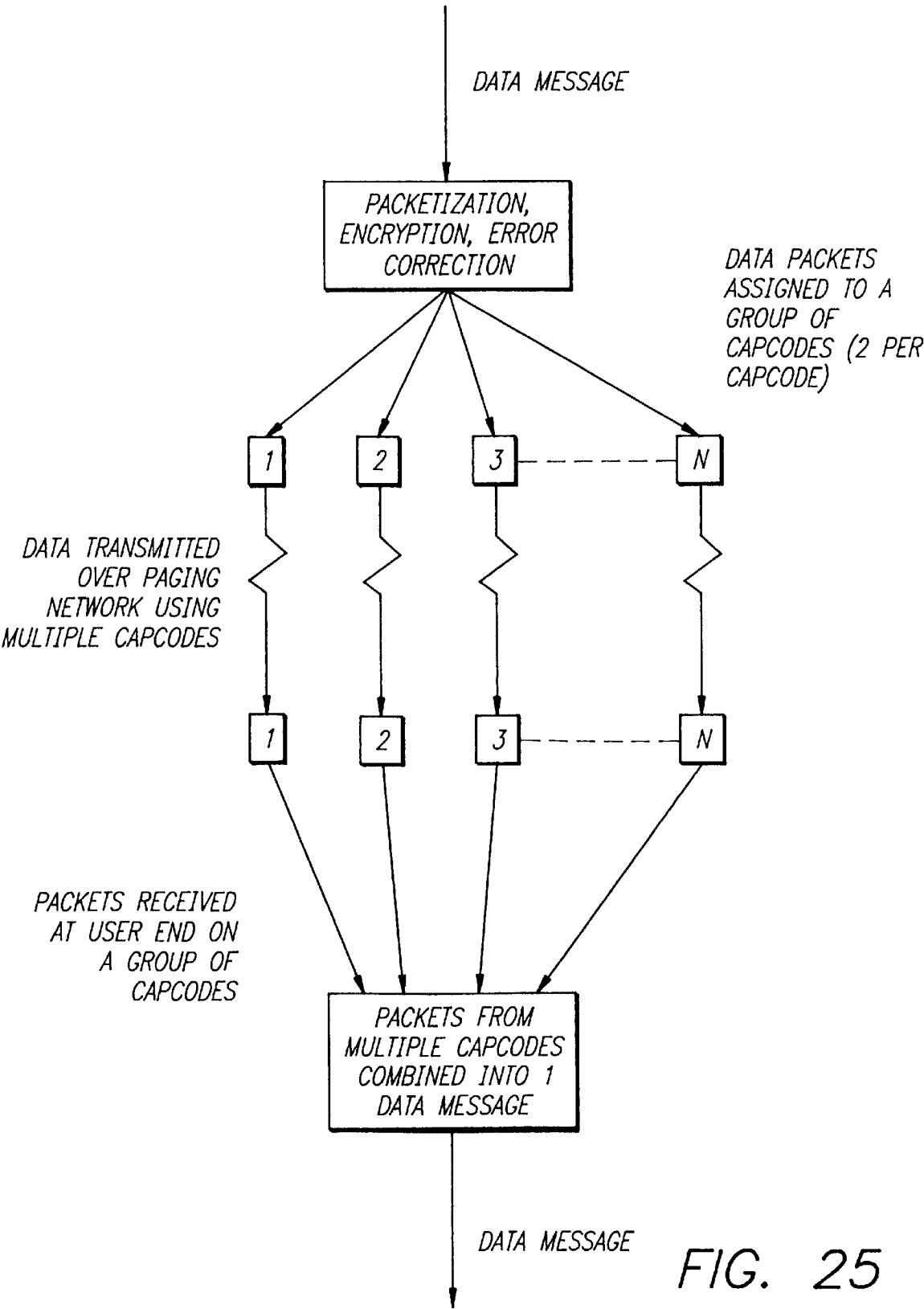


FIG. 25

6,021,433

1

**SYSTEM AND METHOD FOR
TRANSMISSION OF DATA**

REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application Ser. No. 60/010,651, filed on Jan. 26, 1996; U.S. Provisional Application Ser. No. 60/014,341, filed on Mar. 29, 1996; U.S. Provisional Application Ser. No. 60/014,735, filed on Apr. 1, 1996; and U.S. Provisional Application Ser. No. 60/026,471, filed on Sep. 23, 1996.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to communication systems, and more particularly to both wired and non-wired data transmission communication systems.

2. Description of the Prior Art

Undoubtedly, computers, communications and information are driving forces in society today. The most significant advances in computers, communications and information have been in the areas of multimedia, wireless and on-line services, respectively. Each of these technologies have produced significant benefits and have effected nearly everyone's life in one way or another.

In particular, more than 100 million personal computers are equipped with multimedia hardware and software and nearly every new personal computer manufactured today is shipped with some form of multimedia. Multimedia has made the computer much more than a number crunching, word processing tool. Rather, multimedia has turned the computer into an indispensable educational, entertainment and information tool. By combining the realism of sound, graphics and video, multimedia applications have revolutionized the way individuals work, entertain and stay informed. Multimedia has also helped drive the computer industry to provide tools which can be used by the most novice computer user making computers almost as prevalent in our society as television or radios. Also, multimedia has driven manufacturers to build smaller and more powerful and mobile systems—leading a technological revolution not matched in our history.

Moreover, wireless communication technology has allowed individuals to be notified anywhere and anytime of information. Wherever an individual is, i.e. whether away from the office or in the car, he or she can be informed of information, such as new meeting schedules, dinner plans or even life or death emergencies.

Additionally, on-line services have revolutionized the distribution of information in our society by making available, to individuals throughout the world, endless amounts of information on every subject imaginable. The Internet and on-line services have brought together the world through a linkage of interconnected computer systems which can share information almost instantaneously.

These technologies suffer from numerous disadvantages, however. The benefits of wireless technology have only been utilized for personal messaging offering limited message lengths and have never been utilized as a computer peripheral, limiting the benefit of instant anytime anywhere to personal messages of limited length and value. Consequently, information which is sent is typically old and historic.

Moreover, while popular in education and business markets, multimedia has yet to find widespread application in the consumer market. While valuable in education and

2

business circles, the average home user has little use for sound and full motion video. As the number of information providers continue to expand throughout the world, the amount of time and effort required to find information becomes exponentially longer.

In particular, the interface to on-line services is often difficult and intimidating to novice computer users. As a result, the benefit of this valuable source of information is thus not available to them. For example, despite the wealth of information available, users are required to search through the myriad of information, rather than having the information come to them. Consequently, information is often missed.

Furthermore, immediate notification of information is not available. For example, users who use computer related services, such as electronic mail (E-mail), do not receive instant notification when new mail is received. As a result, urgent E-mail will sit unnoticed in an electronic mailbox.

Another major problem is that data transmitted over existing wireless broadcast networks suffer from inevitable degradation. Traditional paging, being a one-way transmission, can use only forward error correction (FEC) on data packets. Many existing paging networks use Motorola's FLEX TM, POCSAG or other wireless protocol's error correction/detection capabilities. Although these industry standard protocols provide error detection capabilities, many of them are not able to deal with burst errors or errors due to loss of synchronization. Since these protocols cannot correct all possible errors, some of the data packets will arrive with errors or simply get lost. In most cases, truncated packets and lost packets account for the vast majority of errors after decoding.

Similar problems exist with other forms of wireless communication systems as well.

What is needed therefore is a system and method for data transmission, which combines the benefits of multimedia, wireless and wired on-line services while addressing and overcoming their limitations.

SUMMARY OF THE INVENTION

The preceding and other shortcomings of prior art methods and systems are overcome by the present invention which provides a system and method for data communication connecting on-line networks with on-line and off-line computers. In particular, the present system provides for broadcast of up to the minute notification centric information thereby providing an instant call to action for users who are provided with the ability to instantaneously retrieve further detailed information. Throughout the day, various pieces of information happening around the world are currently available in a sender initiated paradigm where individuals have to seek out the information. In accordance with the present invention, the notification centric portions of that information that lives in an electronic medium is wirelessly broadcast on a nationwide basis to wireless receiving devices which are attached to personal computers or other computing devices. Upon receipt of the information at the personal computer, the user is notified through different multimedia alerts that there is an incoming message. Wirelessly broadcasted URL's, associated with the data, are embedded in data packets and provide an automated wired or wireless connection back to the information source for obtaining detailed data.

The present invention unlike other wireless systems provides for a combination of broadcast, narrowcast and point-cast transmission. That is, information can be transmitted

6,021,433

3

wirelessly to everyone (broadcast), to a subset of users (narrow cast) or to one user (pointcast). The present invention furthermore provides multiple viewers which listen to the airwaves and have the ability to filter against the broadcast with specific action. A message server provides different types of filters with the ability to parse data. Additionally, the message server is designed such that third party developers can write different types of multimedia viewers which can easily be downloaded to the user system and automatically registered with the message server. The viewers can thus be controlled through the interface of the present invention and multiple viewers and multiple controllers of such viewers can dynamically be added and controlled. Moreover, since the messages are encoded for multimedia events, the viewers of the present invention have capability to do different things for multimedia, such as sound, video, animation and so forth.

In operation, data parsed from a plurality of incoming data feeds from existing information sources is prepared for optimized wireless transmission and then transmitted nationwide to connected and non-connected computing devices thereby extending the reach of existing information sources, such as Internet and on-line services. On the user end, once data is received, a global communications server recombines, decodes, decrypts and decompresses the incoming data. When a complete data message is formed, the communications server sends a message to the user interface alert panel causing an animated icon to fly to the alert panel notifying a user that a new message has arrived. Upon clicking the icon, the appropriate viewer is launched. Users can then display the context of the data on their computers. Based on preferences set by the user with respect to sound, video and animation, users can be alerted to incoming messages. Wirelessly broadcasted URL's and on-line addresses, associated with the data, are embedded in multimedia viewers and provide an automated wired connection/link back to the information sources to obtain detailed information. Information, such as advertisements and promotional broadcasts, can be embedded in a multimedia viewer as well as automatically activated on a scheduled or triggered basis. Information is thus modified and updated instantaneously and wirelessly. Additional information services can be activated wirelessly through broadcast activation codes which can enable or disable services.

The present invention also provides a method based on Reed-Solomon code which is used to derive redundant data packets thereby minimizing redundancy, and maximizing flexibility and packet recovery ability.

In accordance with another embodiment of the invention, the information provided from the information sources and transmitted to the central broadcast server to be consolidated in accordance with the present invention and then transmitted wirelessly nationwide to personal computers and other computing devices can also be sent simultaneously via a wired connection to the same personal computers and computing devices having Internet/World Wide Web (WWW) access (direct or via on-line service providing Internet and Web access).

The foregoing and additional features and advantages of this invention will become apparent from the detailed description and accompanying drawing figures that follow. In the figures and written description, numerals indicate the various features of the invention, like numerals referring to like features throughout for both the drawing figures and the written description.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is schematic diagram of a wireless communication network including information mirroring, selection

4

addressing, bandwidth optimization, message server design and URL broadcast and hotlinks in accordance with the present invention;

FIG. 2 is a block diagram of the wireless communication network illustrated in FIG. 1;

FIG. 3(a) is a block diagram of the head-end high-level software architecture for communication over a paging network in accordance with the present invention;

FIG. 3(b) is a block diagram of the head-end high-level software architecture for communication over a Vertical Blanking Interval (VBI) in accordance with the present invention;

FIG. 3(c) is a block diagram of the head-end high-level software architecture for communication via satellite in accordance with the present invention;

FIG. 4 is a flow chart illustrating the transfer of data from the content manager to the wireless broadcast network;

FIGS. 5-1 and 5-2 is a table illustrating the 8-bit binary format for information notification data blocks;

FIG. 6 is a table illustrating the 8-bit binary format for personal alert notification data blocks;

FIG. 7 is a table illustrating the 8-bit binary format for messages;

FIG. 8 is a table illustrating the 8-bit binary format for packets;

FIG. 9 is a table illustrating the 8-bit binary format for single packet data blocks;

FIG. 10 is a detailed schematic diagram of the message server design illustrated in FIG. 1;

FIG. 11 is an illustration of a user remote interface for controlling the computer interface in accordance with the present invention;

FIG. 12 is a flow chart of an algorithm for extracting and processing the Internet source URL for messages broadcast over the wireless communication network illustrated in FIG. 1;

FIG. 13 is a flow chart of an algorithm for generating and processing E-mail alerts in accordance with the present invention;

FIG. 14 is a flow chart of an algorithm for address and message filtering in accordance with the present invention;

FIG. 15 is a detailed flow chart of the algorithm illustrated in FIG. 14 for targeting data to a user utilizing physical and virtual addresses;

FIG. 16 is an illustration of the columns of a data group encoded by an encoder using a modified Reed-Solomon code for deriving parity-check packets;

FIG. 17 is a flow chart of an algorithm for deriving parity-check packets as illustrated in FIG. 16;

FIG. 18(a) is a flow chart of an algorithm for data compression which combines Huffman compression and dictionary-based compression in accordance with the present invention;

FIG. 18(b) is a flow chart of an algorithm for data decompression of the compression algorithm illustrated in FIG. 18(a);

FIG. 19(a) is a flow chart of an algorithm for data compression using differencing in accordance with the present invention;

FIG. 19(b) is a flow chart of an algorithm for data decompression of the compression algorithm illustrated in FIG. 19(a);

FIG. 20 is an illustration of a user interface alert panel as seen by a user;

6,021,433

5

FIG. 21 is a flow chart of an algorithm for implementing the initialization procedure for the user interface alert panel illustrated in FIG. 20;

FIG. 22 is a flow chart of the algorithm for implementing process EMIT messages procedure for the user interface alert panel;

FIG. 23 is a block diagram illustrating how star feed messages are processed in accordance with the present invention; and

FIG. 24(a) is a depiction of a market scoreboard viewer;

FIG. 24(b) is a depiction of a football viewer;

FIG. 24(c) is a depiction of a newspaper viewer;

FIG. 24(d) is a depiction of a stock ticker viewer; and

FIG. 25 is a flow chart of the algorithm for

DETAILED DESCRIPTION OF THE
PREFERRED EMBODIMENT(S)

Referring to FIG. 1, a wireless communication system 10 including selection addressing 28, connecting on-line information sources 12 with on- and off-line computers, such as personal computer 14, is illustrated. In accordance with the present invention, the wireless communication system 10 turns a personal computer 14 or other computing device into a personal wireless information and messaging center. Although the present invention may be used to interact wirelessly with any computing device, for illustrative purposes, the present invention will be described and illustrated utilizing a personal computer 14. One skilled in the art will recognize that computing devices may include consumer electronic devices including computing capabilities. The data/information which is transmitted in accordance with the present invention may be in the form of voice (audio), video, data or a combination thereof.

In particular, the present system provides for broadcast of up to the minute notification centric information thereby providing an instant call to action for users who are provided with the ability to instantaneously retrieve further detailed information. Throughout the day, various pieces of information happening around the world are currently available from information sources 12 in a sender initiated paradigm where users have to seek out the information. In accordance with the present invention, the notification centric portions of that information that lives in an electronic medium is wirelessly broadcast on a nationwide basis to wireless receiving devices 32 which are connected to personal computers 14 or other computing devices. Upon receipt of the information at the personal computer 14, the user is notified through different multimedia viewers 20 that there is an incoming message. The message can be of something that is happening at the present moment anywhere around the world. Included with the broadcast that is wirelessly sent to the user is the Internet address and location of the detail of that message. By clicking on a button within the multimedia viewer 20 that notified the user that a message came in, the present invention will automatically make a wired connection to the information source 12 utilizing the user's preferred on-line browser which will direct the user to the particular location on the Internet service provider where the user can receive detailed information.

The information source 12 may be a private Internet provider such as Quotecom, corporate Internet provider or an on-line service provider such as America On-Line, Compuserve, Prodigy, the Microsoft Network, and the like. A browser is a known software tool used to access the information source 12 via the providers. Known browser

6

software includes Netscape, Netscape Navigator, Microsoft Explorer, Mosaic and the like. The present invention is designed to operate with any of these known or developing web browsers.

Additionally, the present invention unlike other wireless systems provides for a combination of broadcast, narrowcast and pointcast transmission. That is, information can be transmitted from a central broadcast server 34 wirelessly to everyone (broadcast), to a subset of users (narrow cast) or to one user (pointcast). One skilled in the art will recognize that the central broadcast server 34 operates effectively as a network operations center. The present invention furthermore provides multiple viewers 20 which listen to the airwaves and have the ability to filter against the broadcast with specific action. A message server provides different types of filters with the ability to parse data. The filters control which messages are handled by a particular viewer 20. Additionally, the message server is designed such that third party developers can write different types of multimedia viewers 20 which can easily be downloaded to the user system and automatically registered with the message server. The viewers can thus be controlled through the interface of the present invention and multiple viewers 20 and multiple controllers of such viewers can dynamically be added and controlled. Moreover, since the signals are encoded for multimedia events, the viewers 20 of the present invention have capability to utilize multimedia capability.

As will be described in detail below, data parsed from a plurality of incoming data feeds 16 from existing information sources 12 is wirelessly transmitted by the central broadcast server 34 nationwide through a commercial wireless carrier 36 to connected and non-connected computing devices 14 thereby extending the reach of existing information sources 12, such as Internet and on-line services. On the user end, once data is received, the message server design 18 recombines, decodes, and decompresses the incoming data. When a complete data message is formed, a communications server 38 in the message server design 18 notifies a user interface alert panel 50 which presents an icon, which when clicked, notifies appropriate viewers 20 which are registered to display particular data. Users can then display the context of the data on their computers 14. Based on preferences set by the user with respect to sound, video and animation, users can be alerted to incoming messages. Wirelessly broadcasted Uniform Resource Locator's (URL's) 22, associated with the data, are embedded in multimedia data packets and provide an automated wired or wireless connection or link 22 back to the information source 12 for obtaining detailed data. A network path to an information source 12 is identified by the URL having a known syntax for defining a network. Data, such as advertisements and promotional broadcasts, can thus be embedded in a multimedia viewer as well as automatically activated on a scheduled or triggered event. Moreover, an advantage of the present invention is that data can be modified and updated instantaneously and wirelessly. Additional services can be activated wirelessly and existing services disabled through broadcast activation codes which can enable or disable addresses thus turning services on and off.

Another advantage of the present invention is that a remote computer 14 can receive information instantly—even while it is off-line (i.e. not connected to the Internet or some other on-line service). Thus, a user has the ability to receive "on-line" information even when the user is "off-line". In accordance with another advantage of the present invention, a user can simultaneously, using the same com-

6,021,433

7

puter 14, work on a conventional application, such as a spreadsheet or word processing program, and monitor information which is being transmitted wirelessly.

The user computer 14 of the present invention includes a microprocessor connected to a system bus and supported by read only memory (ROM) and random access memory (RAM) which are also coupled to the system bus. The RAM is the main memory into which the operating system and application programs are loaded. The RAM may also support Internet services, including but not limited to the file transfer protocol (FTP) and simple mail transfer protocol (SMTP) or E-mail. A CD ROM, which is optional, is connected to the system bus and is used to store a large amount of data. Various I/O controllers, including but not limited to the video controller, audio controller and mouse controller may also be connected to the system bus. A modem enables communication over a network to other information sources or computers. The operating system of the computer may be Windows '95 (TM), WINDOWS NT (TM) or any other known and available operating system.

In the preferred embodiment of the invention, the user computer has a 486 PC or higher processor, 16 MB of RAM, Windows 95 operating system, at least 20 MB available on hard disk for storing the executable programs, support files and databases, sound and video cards, monitor, mouse or other equivalent pointing device, an ISA slot for receiving an internal 16 Bit ISA receiver card, or serial port. The receiver card installed in the ISA slot in the user computer 14 interacts with the wireless receiver 32. The wireless receiver may also be accessed via the serial port. One skilled in the art will recognize that the present invention is not limited to the particular configuration discussed above. Rather, the present invention may be implemented on other computer systems and configurations, including but not limited to Macintosh or Unix computers, televisions, telephones, appliances and so forth.

The wireless communication system 10 of the present invention includes information mirroring 26, selection addressing 28, bandwidth optimization 30, receiving means 32, message server design 18 and URL broadcast and hot links 22.

Information Mirroring

As is illustrated in FIG. 1, information sources 12, such as the Internet, on-line services and other information sources, provide data feeds, including real time data feeds, to a network of servers 33 in the central broadcast server 34. These data feeds, once they have been parsed, compressed, encrypted and packetized based on feed and data type, provide the basis for outgoing broadcast sent immediately or on a scheduled basis. The data feeds include but are not limited to, electronic mail (E-mail) and other personal alert notifications, news, sports, and financial stories, premium and special event feeds, advertisements/promotions, graphics, sounds, and scheduled updates. The data feeds generated by the information sources 12 are in digital form and divided into one or more data packets.

Referring to FIG. 2, a block diagram 100 of the software architecture for communications between the information sources 12 and central broadcast server 34 prior to transmission to users is illustrated. Referring to FIGS. 1 and 2, information sources 12 provide data feeds to the central broadcast server 34 which performs selection, scheduling and addressing 28. In particular, real time data feeds from the Internet 13 in the information source 12 are provided to a network of servers 33 in the central broadcast server 34, such as the FTP server 102 and the SMTP server 104 illustrated in FIG. 2. The data, which can include but is not

8

limited to stock quotes, weather, lotto, E-mail, etc. is then respectively parsed by parsers, such as the stock quote parser 106, weather parser 108, lotto parser 110 and mail parser 112, and then transmitted to the content manager 114 located in the central broadcast server 34. Data is also provided to the central broadcast server 34 by sources 116 which provide software and hardware for a mainstream connection, via FM radio, with the source 118. This kind of data is also parsed by various parsers, such as Reuters 120, COMDEX 122 and TSN 126. The present invention is not limited to the information sources or parsers described herein. Rather, any type of information source and corresponding parser may be used. The parsed data is then transmitted to the content manager 114.

The central broadcast server 34 also provides a registration/subscription processor 128 via the World Wide Web (WWW) database or alternatively, other means. The WWW is a collection of servers of the Internet that utilizes the Hypertext Transfer Protocol (HTTP). Through the registration/subscription processor 112, a user can register and subscribe to receive broadcasts provided by the present invention via the user computer 14. The information provided by the user is transmitted to a subscriber database 130 which is utilized by the central broadcast server to determine which subscribers receive which types of content.

Referring to FIG. 2, the content manager 114 determines how different types of information are handled. In particular, it specifies priorities for different types of information, and decides which pieces of information will be transmitted and which will be rejected. It also applies scheduling rules 132 to determine when messages should be scheduled to be transmitted to the user. In addition, the content manager 114 is responsible for determining what format the information should be sent in, what compression method to use, and who information should be sent to. The compression method and format are determined by the type of information. When and if the information should be sent, who it should be sent to, and the priority of the information are determined based on the type of information, the time of day, the day of the week, and the specific date. So, for example, these rules could be used to specify that certain news feeds go to premium subscribers only except during certain hours of the day. Or it could be used to say that stock quotes are a low priority during hours the stock exchanges are closed, on Saturday and Sunday, and on market holidays. The content manager 114 also has the ability to detect and remove duplicate messages.

The content manager 114 communicates with the information gateway 134 which is responsible for resolving logical information inside the system to physical information needed for the wireless gateway 136. In particular, the information gateway's 114 duties include, but are not limited to: resolving service identifications (ids) and addresses from a logical address and managing the content budget rules 138 to ensure that the total content quota is not exceeded. The content budget is based on the number of bytes which may be transmitted in an hour. The algorithm used manages the budget by evaluating the total bytes allowed in the hour, the priority of the information, the total bytes sent so far in the hour and the maximum instantaneous rate at which information may be sent to determine whether to send a message. The goal being to ensure that sending low priority information early in the hour will not prevent high priority information from being sent late in the hour. Since the input to the information gateway 134 is primarily logical, it could be exchanged for an information gateway 134 which could send the information to be transmitted over another

medium, such as the Internet. In addition, the information gateway 134 enforces priorities to ensure that higher priority information is sent before lower priority information.

In accordance with the present invention, the wireless gateway 136 prepares data blocks for transmission over a wireless broadcast network, including but not limited to transmission via a paging network (FIG. 3(a)), Vertical Blanking Interval (VBI) (FIG. 3(b)) or satellite (FIG. 3(c)), narrow and broadband PCS, GSM, VSB television, cellular and other developing wireless technologies. One skilled in the art will recognize that the data blocks can be transmitted by a digital, analog or FM subcarrier. The present invention is designed to operate with any of the above known or developing transmission networks.

In particular, referring to FIG. 3(a), a block diagram of the head-end high-level software architecture for transmission over a paging network 37 in accordance with the present invention is illustrated. The paging network 37 allows information to be transmitted over paging frequencies to paging receivers 32 which are connected to a user computer 14. The wireless gateway 136 transmits information to a plurality of paging terminals 39 which transmit the information to paging transmitters 41. In turn, the paging transmitters 41 transmit the information to receivers 32, which only receive information having specific addresses as noted in detail below. The paging terminals 39 and transmitters 41 are preferably located nationwide to provide information access to all users. Paging terminals communicate with one another via the Inter and Intra System Protocol (TNPP). Information is typically received at a paging terminal 39 and eventually transmitted to a separate paging transmitter 41 through a radio control link. One skilled in the art will recognize that the link between the paging terminal 39 and the radio controlled link to the paging transmitters 41 can be a satellite link. In particular, information from the paging terminal 39 is transmitted to a satellite via an uplink. The information is then modulated onto the carrier of the radio control link for transmission to the paging transmitters 41. One skilled in the art will recognize that any commercial paging carrier which can transmit information wirelessly can be utilized in accordance with the present invention.

Referring to FIG. 25, in accordance with an advantage of the present invention, to overcome the paging network limitation on the amount of data that may be sent to a single address, or capcode in paging terminology, messages are sent on groups of pooled addresses and received at the user end on corresponding pools of addresses. Thus, information is multiplexed over multiple addresses but is reassembled at the user end as if sent to a single address. This allows utilization of available network bandwidth that could not be utilized with a single address.

In particular, the data to be transmitted over a paging network 37, such as that illustrated in FIG. 3(a), first goes through a process of packetization, encryption, compression and forward error correction methods, as described in detail below. The output of this process is 1 to n number of data packets, depending on the level of error correction, and type of compression/encryption applied to the data. The paging network addresses an individual or group by broadcasting on a particular address or capcode. By programming a paging device to listen to the individual capcode, the device is then capable of receiving the particular message. The inherent problem with the FLEX protocol which is used by major paging carriers is that there is a limit to the number of messages which can be sent to any one particular capcode at a time. In accordance with FLEX encoding rules, only 2 messages per capcode can exist at any one time in a

particular FLEX frame, which is approximately 1.875 seconds. A typical data message sent over a paging carrier is broken down into 16 individual data packets. If only one capcode is transmitted, it would take $(16 \text{ packets/message}) * (\frac{1}{2} \text{ frame/packet}) * (1.875 \text{ sec/frame}) = 15 \text{ seconds/message}$. This is a relatively slow rate and only utilizes a small fraction of the FLEX frame. A FLEX frame is capable of transmitting on four different phases or channels at a particular time, hosting several messages per frame. The FLEX encoding rules only specify the maximum messages per capcode frame, but there is no limit set to the number of capcodes.

Referring to FIG. 25, in accordance with an advantage of the present invention, the data message is multiplexed over a number of capcodes (i.e. uses multiple capcodes to send one message). Using the previous example, the present invention would send the 16 packets of the data message to 8 different capcodes. Thus, it would take $(16 \text{ packets/message}) * (\frac{1}{2} \text{ capcodes/message}) * (\frac{1}{8} \text{ frame/capcode}) * (1.875 \text{ sec/frame}) = 1.875 \text{ sec/message}$. The data rate is approximately 8 times faster and fully utilizes the FLEX frame. Although the relationship between the capcode and the packet id number is arbitrary, the server software assigns the packets in a "round-robin" fashion, assigning packets 1-8 to capcodes 1-8, respectively, and packets 9-16 to capcodes 1-8, respectively.

At the user end, the software decodes the messages in a similar manner. A user would subscribe to a particular service, which essentially translates into a set of capcodes which are programmed into the receiving device 32 (FIG. 3(a)). The receiving device 32 then receives the packets which are transmitted to that particular set of capcodes. Thus, for example, the user software would initialize the receiving device 32 with the same 8 capcodes as on the transmit side. The packets received with those 8 capcodes would then be combined into the original data message.

Referring to FIG. 3(b), a block diagram of the head-end head-level software architecture for transmitting data over a Vertical Blanking Interval (VBI) of a television signal 135 in accordance with the present invention is illustrated. The wireless gateway 136 transmits information through a standard RS232 interface 137 and modem 139, which through a telephone line 141 communicates with a modem 143 at a television network broadcast transmission site. The information is forwarded from the modem 139 to a VBI encoder 145 which combines the VBI data with a standard television video signal 153. The encoded data is then forwarded to a satellite uplink transmitter 147 which transmits the television signal 153 to a satellite antenna/receiver 151 via satellite 149. A VBI decoder 155 then extracts the data from the television video signal and performs physical device addressing. The VBI encoder and decoder may be any commercially available encoder and decoder designed for VBI transmission. The communications server 38 is modified to interface with the driver for the VBI decoder 155 which is provided by the manufacturer of the decoder hardware.

Referring to FIG. 3(c), a block diagram of the head-end high-level software architecture for transmission via a satellite-based system 157 in accordance with the present invention is illustrated. The wireless gateway 136 transmits information through a standard RS232 interface 159 and modem 161, which through a telephone line 163 communicates with a satellite modem 165. The information is forwarded from the satellite modem 165 to an uplink transmitter 167 which transmits the data to a satellite dish or antenna 171 via satellite 169. In particular, the satellite dish

6,021,433

11

or antenna 171 receives the RF signal from the satellite 169. A standard satellite receiver PC card 32 converts the RF signal into PC compatible data. The communications server 38 is modified to interface with the receiver card driver provided by the manufacturer of the receiver PC card 32 to receive data from a standard satellite data receiver.

The content manager 114 utilizes a content programming station 140 to control the content of programming. The content programming station 140 allows a programming manager (not shown) to alter the rules used by the content manager 114. The content programming station 140 will also be used to review and alter content schedules and schedule ad hoc messages. For example, if there are news feeds which must be manually filtered to locate acceptable content, the news feeds would appear at the content programming station 140 for the program manager to review.

A flowchart illustrating the algorithm for implementing the processing of data prior to transmission is illustrated in FIG. 4. Information from the content manager is initially applied to the information gateway 134 (step 115) which resolves its logical destination address to a physical wireless address based on information in the subscriber database (step 117). The data is then applied to the wireless gateway 136 which creates the data block, performs packetization, compression, encryption, and so forth to prepare the data block for transmission over the wireless broadcast network (step 119). The data block is then transmitted over the wireless broadcast network by the commercial carrier 26.

Information Mirroring

Data is transmitted from an information source to the central broadcast server 34 as discrete message blocks using E-mail or a well-known high speed protocol such as the Transport Control Protocol/Internet Protocol (TCP/IP). (See Comer, D.E., "Internetworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture, Second Edition", Prentice Hall, Englewood Cliffs, N.J. (1991).) In particular, each data packet transmitted by the information source 12 includes a header, packet data and information to ensure proper transmission to the central broadcast server 34. Additionally, an error correction code is typically added to each packet prior to transmission. The data block is broken down into messages and messages are broken into packets. Each packet is accompanied by a message id and a sequence number. All packets belonging to the same message contain the same message id. A sequence number denotes the position of the packet inside the group. Some packets will also carry the total number of packets belonging to the message. Each packet header includes the following: packet type (4 bits), total packets included (1 bit), message identifier (11 bits) and packet sequence number (1 byte).

Although the preferred transmission protocol from information source to the central broadcast server 34 is TCP/IP, it will be appreciated by those skilled in the art that many other standard or application specific protocols, such as the Open Systems Connection (OSI), may be used as well.

The information sources 12 thus provide the information basis for outgoing broadcast transmitted by the central broadcast server 34 through nationwide wireless broadcast network immediately or on a scheduled basis to both on- and off-line computers 14. When the central broadcast server 34 receives the data packets from the information source 12, it pre-processes the data packets and wirelessly transmits the data packets to both on- and off-line computers 14. Consequently, computer users receive real time notifications of information, including but not limited to breaking headlines, sport scores, weather disasters, financial information and even the arrival of new electronic mail. It will be

12

understood by one skilled in the art that the information consolidated at the central broadcast server 34 may additionally be sent via a wired connection to a personal computer or computing device.

Referring to FIG. 1, information sources 12 also receive requests from remote personal computers 14 or other computing devices for more detailed information. Wirelessly transmitted URL's 22, associated with incoming information, are embedded in the broadcast message from the central broadcast server 34, which is displayed in the multimedia viewers 20 and provide an automated direct wired or wireless line connection 22 back to the information source 12 such that detailed data may be automatically downloaded to the user's computer 14.

As illustrated in FIG. 1, data generated by the information sources 12 is fed to the central broadcast server 34, which processes the incoming data packets by parsing the feeds 16 against specific filters, encoding the data and creating desired broadcast feeds for wireless transmission as described in detail below.

Selection Addressing

As is illustrated in FIG. 1, the data packets generated by the information sources 12 are transmitted to the central broadcast server 34, where they are internally processed before being wirelessly transmitted through a carrier 36 to one or more personal computers 14 or other computing sources via selective receivers 32. When the packets arrive at a user receiver 32, they are reassembled by the communications server 38 in the message server design 18 into the original message. One skilled in the art will recognize that the carrier can be a local, regional, nationwide or worldwide carrier.

Information from the content providers is first formatted according to the proprietary EMIT protocol before being prepared for transmission over the wireless broadcast network. In the EMIT format, information feeds include a number of parts, each separated by the tilde (~) character. Each part begins with a tag (keyword) followed by an equal sign (=) and the data for that part. The tag determines how to interpret the data in that part. Most tags are single characters to minimize network traffic. Also, tags are case sensitive to allow more single character tags. Tags 1-5 are reserved for information category and sub categories. Other tags generally are derived from the first character in a name, such as, H for headline. An example of an EMIT format information feed is provided below: 1=S~2=B~H=Dodgers Win World Series~D=Nov. 02, 1989 9:30 pm where the primary category (1=) is S (which stands for sports), the first sub category (2=) is B (which stands for baseball), the news headline (H=) associated with this feed is Dodgers Win World Series, and the date/time (D=) is Nov. 02, 1989 9:30 pm.

Data from the information sources is packed into 8-bit binary format data blocks in the central broadcast server 34. The two basic data block types are illustrated in FIGS. 5 and 6. In particular, FIGS. 5-1 and 5-2 defines the 8-bit binary format for "information" notification data blocks while FIG. 6 defines the 8-bit binary format for "personal alert" notification data blocks. Information notification data blocks, illustrated in FIGS. 5-1 and 5-2, contain general information targeted to all users, including but not limited to news headlines and stories, sports scores, financial market data, and so forth. Personal alert notifications, illustrated in FIG. 6, contain alert information targeted to specific users, including but not limited to notifications regarding E-mail arrival, stock prices reaching specified values, Internet telephone calls, chats or meeting notices.

6,021,433

13

Prior to transmission, at the central broadcast server **34**, the data packets are encoded using a protocol suitable for the transmission of information. Data blocks are packetized for transmission over the wireless broadcast network using transmission protocols.

In the preferred embodiment, which uses the paging network as the means of wireless broadcast or transmission, Motorola's FLEX (TM) protocol is utilized. Alternatively, other protocols, such as traditional Post Office Code Standardization Advisory Group (POCSAG) protocol, Motorola's REFLEX TM and INFLEXION TM, AT&T's protocol derived from CDPD or other developing protocols may be used as well. Most wireless transmission protocols, including POCSAG, provide random error correction as well as error detection capabilities, thereby adding error detection and correction capabilities to the information link.

Depending on the type and amount of information contained, a data block may be enclosed in a single packet, or parceled into messages which in turn are subdivided into one or more packets. The message format protocol is illustrated in FIG. 7. Large data blocks are divided into messages for efficiency in transmission. The data block header is sent as part of the message. The header type item is used to distinguish between the data block and message headers.

The basic unit of transmission is the packet. Each packet includes a header and contents. The information contained in the header defines the packet's contents. In accordance with the present invention and as illustrated in FIGS. 8 and 9, two basic types of packets in the 8-bit binary format are utilized. The first 4 bits in the packet define the packet type. Standard packets are used for transmitting data blocks too large for a single packet. In this case, each packet contains the ID of the message to which it belongs, and the packet number denoting the position of the packet inside the message. This allows the software at the user receiving end to rebuild the original messages and data block from the individual packets. Prior to transmitting the packets in a message, forward error correction packets are added as described in detail below. The single packet data block is used where the complete data block can fit into one packet. In this case, the packet header is followed by the data block header and data block contents. Binary alert packets are a special case of the single packet data block and are reserved for the predefined alert notifications described above.

At the receiving end, as described in detail below, the reverse of the data packetization process described above occurs. In the case of multiple packet data blocks, individual packets are combined to form messages based on packet sequence number and message ID included in the packet header. Error correction is performed as required. Individual messages are then combined to form data blocks based on message sequence number and data block ID in the message header.

The central broadcast server **34** performs the following processes on the incoming data: compression, forward error correction, encryption, packetization and wireless broadcast format encoding. After internal processing, the formatted data packets are queued for wireless transmission to their respective destinations which could include one or more remote personal computers **14** or computing devices. In accordance with the present invention, the formatted data packets are either immediately wirelessly transmitted to their respective destinations or stored into available memory for subsequent wireless transmission to their respective destinations. For the latter, i.e. delayed transmission, the central broadcast server **34** includes a non-volatile storage medium for longer term storage of data programmed for subsequent wireless transmission to one or more users.

14

a. Encryption

To minimize unauthorized use of broadcast data, the data is encrypted prior to wireless transmission so that anyone surreptitiously coming into possession of the data would not be able to convert the data to clear form for use. The user software is designed such that it can properly decrypt the data once it is received on the user end. In the preferred embodiment, data is encrypted using the Data Encryption Standard (DES) algorithm. (See "Data Encryption Standard", Federal Information Processing Standards Publication No. 46, January 1977; "DES Modes of Operation", Federal Information Processing Standards Publication No. 81, December 1980.) Alternatively, other known reversible encryption algorithms may be used for data encryption.

Prior to transmission, the data is also encoded with a data signature. The National Institute of Standards in Technology (NIST) Digital Signature Standard (DSS) algorithm is preferably used for signature verification. Alternatively, other known methods of signature verification may be used. (See "Announcing a Digital Signature Standard", Federal Information Processing Standards Publication, Draft 19 Aug. 1991, front page and pp. 1-4; "Specifications for a Digital Signature Standard (DSS)", Federal Information Processing Standards Publication, Draft 19 Aug. 1991, pp. 1-11.) In operation, DSS is used to authenticate the origin of the data (i.e., establish the identity of the signer) and to check the integrity of the data (i.e., confirm that the data has not been altered after it has been signed).

b. Forward Error Correction

To compensate for transmission errors during wireless broadcast, forward error correction algorithms, such as Fire Codes and various forms of Reed-Solomon Codes, are applied to the outgoing data packets. Reed-Solomon and other coding systems are discussed in, for example, Theory and Practice of Error Control Codes, Richard E. Blahut, Addison Wesley, 1983, at pages 174 and 175. A feature of the forward error correction used here is that the ideal packet size is dynamically computed so as to minimize total over the air size while maximizing error correcting capability.

c. Derivation of Redundant Data Packets

Referring to FIGS. 16 and 17, as shown in detail below, the columns of a data group **150** are encoded by an encoder using a Reed-Solomon (RS) code for deriving parity-check packets **152** i.e. redundant packets. In accordance with the present invention, the RS code, conventionally used for error detection and correction, is utilized in a novel manner with respect to reconstructing packets that arrived with errors. As described in detail above, the data transmission in the present invention is based on a wireless protocol, such as Motorola's FLEX TM protocol or the POCSAG protocol which provides error detection capabilities. However, these protocols cannot compensate for burst errors or errors due to loss of synchronization, which often results in truncated or lost packets at the receiver. In the present invention, each information packet **154** which arrives with an error or errors is considered a lost packet. Therefore, an information packet **154** either arrives without error or is lost.

The present invention is thus directed to compensating for such truncated or lost information packets by sending redundant packets. Instead of sending each packet twice or thrice, the present invention utilizes a modified RS code in a novel manner to transmit packets with redundancy as explained in detail below. For example, for a message which is split into 200 information packets sent over a paging network with a packet loss rate of 1%, the probability of a successful reconstruction of the message is only approximately 13.4%. If every information packet is sent twice, i.e. 400 total

6,021,433

15

packets, the probability of a successful reconstruction of the message increases to approximately 98.2%. In accordance with an advantage of the present invention, by using a modified RS code to derive redundant packets, only 5 extra packets, i.e. 205 total packets, need to be sent to achieve the same approximate 98.2% successful reconstruction probability. Thus, the present invention provides an improvement over conventional methods, which utilize additional error correction and detection capabilities on a per packet basis. In the present invention, Reed Solomon parity check packets 152 effectively compensate for lost information packets. As a result, redundancy and packet loss rate are minimized, and flexibility and packet recovery rate are maximized.

In accordance with the present invention, data received from an information source is encoded into data blocks at the broadcast server. Each data block is then parceled into one or more messages so that each message can be parceled into information packets 154. Each data packet is accompanied by a message identifier and a sequence number. As described in detail above, all packets which belong to the same message contain the same message identifier. The sequence number denotes the position of the data packet inside the message. Some packets will also be accompanied by information regarding the total number of packets belonging to a message. When enough packets arrive at the user receiver 32, they will be reassembled into the original message by the communications server 38 in the message server design 18 as explained in detail below.

Referring to FIG. 16, in accordance with the present invention, a Reed Solomon code is computed down the columns of the block of data packets, thereby creating Reed Solomon parity-check packets. The most general case (n,k) is adopted where

1 ≤ n ≤ 255 (1)

1 ≤ k ≤ n (2)

where k=number of information packets generated by parceling the input message,
n=total number of transmitted packets.

The total number of transmitted packets is determined based on the degree of protection requested. By allowing for the arbitrary combination of n and k, maximal flexibility is achieved. In particular, n and k are chosen during run-time, instead of design-time. For example, (255,223), (255,251), (7,3), (16,1) Reed Solomon codes, used column-wise are all possible combinations for generating Reed Solomon parity-check packets. In a typical operation, by using a (255, 223) Reed Solomon code column-wise, 32 parity-check packets are generated for a group of 200 information packets to be transmitted. Thus, even if 32 arbitrary packets out of 232 total data packets were lost during transmission, a successful reassembling of the information can still be achieved at the receiver end.

In accordance with the present invention, to minimize the number of lost messages, the information packets are sent with redundancy using a method based on Reed-Solomon code to derive Reed Solomon parity-check packets. Utilizing an 8-bit Reed-Solomon code, the maximum number of data packets (including both information packets and Reed-Solomon parity-check packets) is 255. There is no limitation on the number of symbols in each data packet as long as they are acceptable by the wireless carrier.

In accordance with the present invention, the modified RS code encodes the data over a Galois Field GF(2⁸) (hereinafter GF(256)) whose field elements are represented by their coordinates with respect to the canonical basis {1,a,a², . . . ,a⁷} where a is a root of the primitive monic polynomial:

16

f(x)=x⁸+x⁴+x³+x²+1(3)

Parity-check packets are generated by encoding k data packets column-wise in accordance with the following generating polynomial g(x) equation:

g(x) = ∏_{i=1}^p (x + aⁱ) (4)

where g(x)=generating polynomial
a=primitive element of GF(256)
p=number of parity check packets
Multiplication and inversion in GF(256) are implemented by table lookup or by algorithm depending on performance requirements.

In the preferred embodiment, the encoder for encoding k data packets column-wise is a software simulation of polynomial division using linear feedback shift register (LFSR), with n and k being changeable. The coefficients of the generator polynomial g(x) are saved in the order of ascending power. Alternatively, the LFSR may be implemented in hardware, with n and k fixed. (See William Wesley Peterson, "Error Correcting Codes", Edition One, pg. 150.)

A series of data packets including both information packets and parity-check packets are formed. The number of symbols in each data packet is limited only by the wireless broadcast system. In accordance with the present invention, no extra error correction is added to each data packet.

The number of parity-check packets, n-k, must be in the range [1, 254] and the number of erasures, i.e. errors whose locations are known, must be in the range [0, n-k]. The erasure locations must be all distinct and sorted in ascending order. In the present invention, RS error correction is performed on each column. Each error in the column corresponds to a lost packet. Since it is known which packet is lost, the locations of all errors prior to RS decoding are known. Thus, in accordance with an advantage of the present invention, the location of the errors is known before RS decoding, thereby providing for maximal error correction. In contrast, conventional applications of RS attempt to find both the magnitude and location of an error.

As shown in FIG. 16, each data packet (including information packets and RS parity-check packets) is parceled into many codewords. The length of each codeword is 32 bits, where 21 bits are for information and 11 bits are for error correction/detection.

The data packets, i.e. information packets and parity-check packets, are then transmitted to the message server unit via the user receiver. FLEX TM provides information regarding whether the packets were correctly received or not. As a result, any error locations are detected prior to applying RS decoding. Decoding is then implemented by syndrome evaluation with known error locations. (See Hasan, Bhargava, and Le-Ngoc, "Reed-Solomon Codes and Their Applications", pg. 79-81.)

In accordance with the present invention, the number of information packets k and the number of Reed-Solomon parity-check packets p can be arbitrarily chosen depending on the transmission condition and the desired accuracy rate. The only condition is that the number of information packets k and the number of parity-check packets together total no more than 255. The restriction

p+k ≤ 255 (5)

is imposed by the use of the finite field GF(256). As stated earlier, each data block will thus first be split into several

6,021,433

17

messages so that each message can be split into k packets that satisfy the above restriction. Up to p packets can be lost without compromising successful reconstruction of the message. In accordance with the present invention, even if some data packets are lost, the full message can be recovered using the redundancy data packets generated by the present invention.

Referring to FIG. 17, a flow chart 160 of the algorithm for deriving RS parity-check packets is illustrated. The data block is initially parceled into one or more incoming messages (step 162), and the messages are then parceled into k information packets 154 (step 164). The number of RS parity-checks packets p is then selected (step 166). The information packets are then encoded column-wise with a modified RS code in accordance the generating polynomial:

$$g(x) = \prod_{i=1}^p (x + a^i)$$

and parity-check packets are generated (step 168). The data packets, which include information packets and RS parity-check packets, are parceled into codewords (step 170). After the data packets have been parceled into codewords, error correction/detection is performed on the codewords (step 172). The data packets are then transmitted to the users (step 174).

At the user end, the number of codewords which have error(s) is counted (step 176). Then it is determined whether each packet has any errors (step 178). If a packet does not have an error, then it is saved (step 180). However, if a packet has one or more errors, it is discarded (step 182) and the present invention waits for more packets (step 188). When there are enough packets (step 184), a message is assembled (step 186). If not, the present invention waits for more packets (step 188). Finally, when there are enough messages, the data block is assembled (step 192).

d. Compression/Bandwidth Optimization

FIG. 18(a) is a flow chart of an algorithm for data compression which combines Huffman compression and dictionary-based compression. In accordance with the present invention, the data blocks are compressed at the central broadcast server 34 end prior to transmission so that maximum amounts of information in compressed or bandwidth reduced form can be transmitted to the selected user or users. As discussed in detail below, at the user end, the data blocks are correspondingly decompressed (FIG. 18(b)).

In the preferred embodiment, the current compression algorithm for English language articles saved in ASCII text format combines the Huffman compression and the dictionary-based compression, such as LZ77 and LZ78 based algorithms. In operation, as the compression algorithm scans the input texts, it not only tries to search for the next item in the previously seen text, but also tries to search for the next item in a static Huffman dictionary, and it chooses a method which produces a better result. After the data is received at the user end, it is correspondingly decompressed.

In particular, referring to the algorithm 200 for implementing data compression in FIG. 18(a), the Huffman dictionary is loaded from the disk storage, the address pointer is positioned to the start of the uncompressed input data in memory and a memory buffer for storing the compressed output data is allocated (step 202). Next, it is determined whether the address pointer is moved to the end of the data input (step 204). If so, bit b=1 is written to the output data and the end-of-data token from the Huffman

18

dictionary is written to the output data (step 206) and the compression routine is done (step 208). If in step 204, it is determined that the address pointer is not at the end of the input data, the compression algorithm scans the input texts, searching for the next item in the previously seen text (step 210) and the static Huffman dictionary (step 212), and chooses the method which produces a better result (step 214).

In particular, in step 210, the data is compressed using the previously seen text. A token T1 is generated by comparing the input data at the input pointer to the previous input data. T1 denotes an index to the previously seen data that has the maximum length match with the current data. L1 correspondingly denotes this maximum length.

In step 212, the data is compressed using the Huffman dictionary which was loaded in step 202. A token T2 is generated by looking for the maximum match of the input data at the input pointer to entries in the Huffman dictionary. T2 denotes an index to the dictionary entry for the maximum match. L2 correspondingly denotes the length of the match.

In step 214, the optimum result (T,L) from (T1,L1) or (T2,L2) is chosen depending on which is larger, L1 or L2. If (T1,L1) is chosen, b is set to 0 (b=0), else b is set to 1 (b=1). b is initially written to the output data followed by the optimal result (T,L). The input data pointer is then advanced by L bytes.

After the data is received at the user end, it is correspondingly decompressed in accordance with the algorithm 220 illustrated in FIG. 18(b). The Huffman dictionary is initially loaded from the disk storage, the address pointer is positioned to the start of the compressed input data in memory and a memory buffer for storing the decompressed output data is allocated (step 222). One bit from the input data is read and saved in b (step 224). Next, it is determined whether b=0 (step 226). If so, the data is decompressed using the previously seen text (step 228). The next token (T,L) is initially retrieved, followed by L bytes of decompressed data from the output buffer at a location denoted by T. The retrieved bytes are denoted by txt, which are then written to the output buffer (step 230). The input data pointer is then advanced by the length of the token (T,L) in bits. The program then returns to step 224 and repeats the steps until the Huffman end-of-token is detected (step 232).

If, in step 226, b is not set to 0, it is determined whether the next token is the Huffman end-of-data token. If so, decompression has been completed (step 234). If not, the data is decompressed using the Huffman dictionary (step 236). The next token (T,L) is retrieved, followed by L bytes of decompressed data from the Huffman dictionary using T as an entry into the dictionary. The retrieved bytes of data are denoted by txt, which as noted previously, is written to the output buffer (step 230). The input data pointer is advanced by the length of the token (T,L) in bits and returns to step 224.

e. Differencing

FIG. 19(a) is a flow chart of an algorithm 240 for data compression utilizing differencing. In accordance with another advantage of the present invention, a differencing algorithm 240 is additionally used to compress the coded data, thereby significantly reducing the number of bytes sent with each transmission. In particular, a dictionary-based compression algorithm, such as LZ77 and LZ78 based compression, can be adapted for this application. File two is described with reference to file one in a minimum number of bytes. In such an algorithm, file one is used as the dictionary.

In particular, the precomputed standard hash table HT for file 1, the dictionary file, is loaded from mass storage (step

6,021,433

19

242). The minimum match length L from the length used in creating the hash table HT and the maximum match length U from the limits on contiguous data block transmission size are set. The memory address pointer to the stream of input data (file 2) to be compressed by differencing with file 1 is retrieved and a memory buffer for the compressed output data is allocated. The algorithm 240 next determines whether the end of the input data has been detected (step 246). If so, the compression is complete (step 248). If not, the hash value H of the next input data substring of length L bytes with the same hashing algorithm used to compute HT is calculated (step 250). The optimal match length ML is then set to 0 and the optimal position MP is set to -1 (step 252). For each position P in HT corresponding to H, the best match length PML at position P in file 1 such that

$$L \leq PML \leq U$$

is determined (step 254). If PML is greater than ML, then ML is set such that $ML = PML$ and MP is set such that $MP = P$. If in step 256, $ML = 0$, the bit value 0 is written to the output buffer (step 258). The byte at the current input buffer pointer is written to the output buffer and the input buffer is advanced by one byte. The algorithm 240 returns to step 246 and continuously iterates until the end of the input data is detected (step 248).

If in step 256, ML is not equal to 0, the bit value 1 is written to the output buffer (step 260). The optimal match length ML and the optimal match position MP are written to the output buffer. The input buffer pointer is then advanced by ML bytes. The algorithm 240 returns to step 246 and continuously iterates until the end of the input data is detected (step 248).

As discussed in detail below, at the user end, the data blocks are correspondingly decompressed in accordance with the algorithm 262 illustrated in FIG. 19(b). The dictionary file, file 1, is initially loaded from mass storage (step 264). The memory address pointer to the stream of compressed input data and retrieved and the memory buffer for the decompressed output data is allocated. It is next determined, whether the end of the input data has been detected (step 266). If so, the decompression routine is complete (step 268). If not, one bit b from the input buffer is read (step 270). It is then determined whether $b = 0$ (step 274). If so, one byte from the input buffer is copied and written to the output buffer. The input buffer pointer is then advanced by one byte. The algorithm 262 returns to step 266 and continuously iterates until the end of the input data is detected (step 268).

If in step 274, b does not equal 0, the match length ML and the match position MP is retrieved from the input buffer (step 278). ML bytes are copied from file 1 at position MP to the output buffer. The input buffer pointer is advanced by the sizes of ML and MP in bytes. The algorithm 262 returns to step 266 and continuously iterates until the end of the input data is detected (step 268).

f. Wireless Data Format Encoding

Where the method of transmission is paging, all outgoing messages are preferably encoded to $\frac{7}{8}$ bit data or true 8 bit data for broadcast over paging networks. After the data is received at the user end, it is correspondingly decoded.

With respect to VBI and satellite transmission, all outgoing messages are preferably encoded to true 8 bit data.

g. Addresses

In accordance with the present invention, outbound data will be segmented and sent to the user by way of the user receiver 32 utilizing common and unique addresses. Addresses are numbers used by wireless receiving devices to

20

identify messages targeted to a user. Addresses are usually stored in programmable read only memory (PROM) in the receiver hardware 32. If the address to which a message is transmitted matches a address stored in the receiver 32, then the receiver 32 will process the message. Otherwise, the message will be ignored. In a typical configuration, general "basic services" are wirelessly transmitted on global common addresses, electronic mail and point-to-point messages are transmitted on personalized or unique addresses, and combined premium services and pay-per-view events are grouped together and transmitted on common addresses. Alternatively, the combined premium services and pay-per-view events may be sent on unique addresses as well.

h. Request for Additional Services

The central broadcast server 34 additionally includes telephone and/or modem interfaces for receiving remote request from users to obtain additional or modify existing services. For example, a user from a personal computer 14 or other computing device, can request additional services or modify existing services by telephoning or modeming the central broadcast server 34, which automatically and wirelessly transmits the new or modified services. Modification of subscribed services may also be performed via the Internet and World Wide Web.

i. Simultaneous Wired Transmission

In accordance with an alternate embodiment of the invention, as explained in detail below, the information provided from the information sources 12 and transmitted to the central broadcast server 34 to be consolidated in accordance with the present invention and then transmitted wirelessly nationwide to personal computers 14 and other computing devices as described in detail above can also be sent simultaneously via a wired connection to the same personal computers 14 and computing devices having Internet/World Wide Web access (direct or via on-line service providing Internet and Web access). In particular, the data processed at the central broadcast server 34, in addition to being transmitted wirelessly, is simultaneously made available through wired connection to a specific web site on the Internet. A user can thus connect to the Web via the Internet and receive information through wired means.

Receiving Means

Referring to FIG. 1, a user receiver 32, connected to a personal computer 14 or computing device, receives wireless transmissions sent by the central broadcast server 34. The user receiver 32 preferably includes an Industry-Standard Architecture (ISA) board with a I C interface to an external wireless receiver and utilizes on-board PCSAG, Motorola's FLEX TM protocol or other wireless receiving device receiving and decoding. In accordance with an advantage of the present invention, Motorola's FLEX TM decoding allows for upgradeability to future receiver protocols without requiring replacement of the internal ISA board. The user receiver 32 also includes an indicator, such as a flashing LED, which indicates reception of incoming messages. As described in detail below, the user receiver 32 includes physical addresses for filtering data prior to being transferred to the personal computer 14. The user receiver 32 may be a specially designed or commercially available receiving unit.

60 Filtering

In accordance with the present invention, filtering of information can be accomplished both at the user receiver 32 and personal computer 14 or computing device. Messages are electronically sent to nationwide and local wireless broadcast networks using both physical and virtual addresses. Physical addresses are tags which reside in the hardware portion in the user receiver 32.

6,021,433

21

In addition to standard physical addresses, the present invention implements a virtual address as illustrated in FIG. 14 and described in detail below. In particular, the virtual addresses reside in the software of the user computer 14. Virtual addresses provide additional filtering of incoming data from the user receiver 32. For example, a message may be received by all receivers 32, but if the message is targeted to a specific virtual address, then only those installations in which that virtual address is activated will process the message. In accordance with an advantage of the present invention, virtual addresses may be activated and deactivated through the broadcast network, allowing for external control over the reception of services in a particular installation. It will be appreciated by those skilled in the art that information filtering can be accomplished utilizing virtual addresses only. Virtual addresses can allow for unlimited filtering of messages on the user end. However, this may increase the resource usage of the personal computer 14. Correspondingly, information filtering can be accomplished by utilizing physical addresses only.

A higher level of filtering based on message category and content is also provided. Users can set various filters based on a variety of preferences at information category or specific content levels to allow for automated filtering of incoming information. At the category level, users can control which categories of information received from the broadcast network are processed and which are discarded. For example, if a user were not interested in sports, all sports information categories, such as baseball, football, golf, etc. can be selected for discarding. At the specific content level, a user can select which subcategories of information within a particular information category will be processed. The user selectable subcategories depend on the type of information contained in that category. Subcategories may include, but are not limited to, source providers for headline news stories, specific industry segments (e.g., electronics, computers, communications, industrial, etc.) for business news, specific teams for sports categories, particular states and games for lottery results, and stocks for which quotes are displayed. For example, a user that wishes to have scores displayed only for baseball games involving the New York Yankees or New York Mets can set the filter for the baseball viewer to discard game results for all teams except those two.

Filtering is accomplished prior to information being transferred to the personal computer's hard drive 14, therefore conserving the personal computer's resources. Referring to FIG. 14, a flow chart of an algorithm for message processing using filtering in accordance with the present invention is illustrated. An incoming message from the central broadcast server end 34 after processing as described above is applied to the receiver hardware 32 (step 200). Physical address filtering in the receiver hardware is then used to determine whether the message should be passed on for further virtual address filtering (step 202). If the message passes physical address filtering, the message is applied to virtual address filtering (step 204). Otherwise, the message is disregarded (step 206). Virtual address filtering is then used to determine whether the message should be passed (step 208) on for further message content filtering (step 210). If not, the message is disregarded (step 212). Message content filtering then determines (step 214) whether the message should be stored in the message database (step 216) for further processing and transmission to the user or disregarded (step 218).

The process of targeting data to an user utilizing real and virtual addresses is illustrated in FIG. 15. Data blocks are

22

built in the information gateway 134 and all applicable real and virtual addresses are determined based on the type of information in the data block and user subscription data from the subscriber database 130. If a data block is to be targeted to a specific virtual address, the virtual address is inserted by the information gateway 134 into the virtual address field of the data block header and the virtual address flag is set. The wireless gateway 136 provides the interface to the wireless transmission network. It prepares data for transmission over the network and implements real addresses in the proper data frames as specified by the standard transmission protocol that is used. At the receiving end, arriving data is first filtered via real addresses in the wireless receiver 32 followed by virtual address filtering in the communications server 38. The communications server 38 first checks the virtual address flag in the data block header. If it is not set, then the data block is passed onto the alert panel 50 for storage and display. If this flag is set, the communications server 38 determines if the virtual address in the data block header matches one in the virtual address database. If there is a match, then the data block is passed onto the alert panel 50. If there is no match, then the data block is discarded.

Message Server Design

Referring to FIGS. 1 and 10, the message server design 18 includes a communications server 38, user interface alert panel 50 and viewer server 58.

a. Driver

As is illustrated in FIG. 10, the driver 44 is preferably a Windows 95 driver for the wireless device hardware 42, although another compatible device may be used as well. The driver 44 provides an interface to access received data and control the hardware 42, as well as inform applications as to the status of the receiver hardware 42.

b. Interface

The interface 46 for the wireless device is preferably an AmFlex DLL 46, although another compatible device may be used as well. The interface 46 is used to pass the data received from the wireless device to the communications server 38 for processing and distribution to other software components. It also provides a means by which the communications server 38 can program the device hardware to receive specific messages and also allows the communications server 38 to determine hardware status.

c. Communications Server

The communications server 38 receives data from the wireless device via the interface 46, extracts the different types of data blocks (messages), passes public data blocks to the user interface alert panel 50 and processes private data blocks locally. The communications server 38 is also responsible for initializing the wireless device and maintaining the address database which determines which received messages will be processed. In addition, it provides diagnostic data on received messages for software debug purposes.

In operation, the communications server 38 is notified of incoming data packets by the driver 44 via the interface 46 through a software callback function. Once data packets are received by the communication server 38, it recombines, decompresses, decrypts, filters via virtual addresses as previously discussed, and error corrects the data packets using techniques corresponding to the processing done at the central broadcast server 34 end. In particular, the communication server 38 initially verifies the integrity of the data packets received using common error correction techniques. After error correction, the data packets are unpacketized and entire messages are assembled. After assembly, the communication server 38 verifies once again that the integrity of the

6,021,433

23

message is maintained. The message is then decrypted using the common password previously established. The data signature on the message is also checked to verify the integrity of the data. The messages are uniquely encoded so that it is known which data packet belongs to which message. The messages are stored in a database and when a complete message is formed, it is transmitted to one or more devices that are registered with the communication server 38. As shown in FIG. 10, the complete message may be transmitted to the user interface alert panel 50, shown in detail in FIGS. 3 and 4 and discussed in detail below. Thus, once the data packets are successfully read off the driver 44, the data is error corrected, decompressed, decrypted and assembled into a complete message. The communications server 38 then notifies the user interface alert panel 50.

d. User Interface Alert Panel

Referring to FIG. 10, the user interface alert panel 50 is the main user interface for the applications software. The user interface alert panel 50, which appears to a user as shown in FIG. 20, is the liaison for messages broadcast from the communications server 38 and delivered to the viewer server 20. The user interface alert panel 50 performs all message archiving to the messages database. The main functions of the user interface alert panel 50 are (i) initialization, (ii) processing EMIT messages, and (iii) timing events. The user interface alert panel 50 is run when the user double clicks on a specific icon or selects the application from a start menu, such as the Windows 95 start menu, and is responsible for other applications, such as launching the communications server 38 and viewer server 20 and passing messages received from the communications server 38 to the viewer server 20. The user interface alert panel 50 also displays "fly-in" graphics and icon buttons to alert the user that a new message has been received, allows the user to open a viewer 48 to examine a received message by clicking on the viewer icon button for that message, and maintains the received messages database. The latter includes saving new messages in the database and deleting old messages after a certain period of time, as explained in detail below. The user also accesses the remote control 54 from the user interface alert panel 50 by clicking a remote control icon.

(I) Initialization

FIG. 21 is a flow chart of an algorithm 300 for implementing the initialization procedure for the user interface alert panel 50 in accordance with the present invention. In step 302, during initialization, the user is prompted for database management (compress the message database). In particular, the user interface alert panel 50 will determine if there are more than a predetermined number of messages written into the database 51. In the preferred embodiment, the predetermined number of messages is 2000+, although one skilled in the art will recognize that any number of messages may be used. If the predetermined number is exceeded, records which have been previously marked for deletion are removed from the database 51. Marked records are typically records which have been read by a viewer and are not targeted for any of the other viewers or applications, yet physically remain in the database. These records are removed when the predetermined number of messages is met, thereby only leaving those records which need to be read.

Following database management, the databases 51 are opened for non-exclusive read/writes (step 304). In accordance with the present invention, the three main databases are the (a) messages database which holds all the messages, (b) SYSAPPS database or systems applications database

24

which holds the viewer specific information such as what is executable, what needs to be run for that viewer to be launched, etc. and (c) V groups database which contains a list of all viewers, their alias names and descriptions.

The next step during initialization involves reading the tool bar initialization information from the registry keys (step 306). In particular, the docking location of the user interface alert panel 50 is determined. The user interface alert panel 50 is dockable at all the corners of the display and can also be floated at the center. The animation defaults are also determined because in the customization for the user interface alert panel 50, the user can turn off the fly-in sequence, buttons animated and/or sound files being played. Which winsock ports need to be used to talk to the communications server 38 and viewer server 20 are also determined at initialization.

The next step is during initialization is to launch the communications server 38 and viewer server 20 (step 308). After the executables for the communications server 38 and viewer server 20 have been launched, the communications server 38 is logged into as a client and the viewer server 20 is logged into as a server such that each knows about the user interface alert panel 50.

Then, buttons are created in the user interface alert panel 50 for messages marked as not read (step 310). For example, some records in the message database 51 are not read because the user closed the user interface alert panel 50 before reading them. In accordance with the present invention, buttons are created on the user interface alert panel 50 for those messages.

Finally, the communications server 38 is queried for valid service plans which include but are not limited to E-mail, premier services and power up services (step 312).

(ii) Process EMIT messages

FIG. 22 is a flow chart of the algorithm for implementing process EMIT messages procedure for the user interface alert panel 50. A message or feed from the communications server 38 via the winsock port is initially applied to the user interface alert panel 50. In step 1, the user interface alert panel 50 determines what feed type is present, i.e. whether the message is a binary, star or EMIT type feed.

A typical binary type feed is an E-mail message. The binary feed is, as discussed in detail below, decompressed into a common EMIT feed and processed as a normal feed.

A typical EMIT type feed involves common user information such as messages for football, scoreboard viewers, horoscope, lottery etc.

A typical star type feed involves a registry value change which creates or updates the appropriate registry key(s). In many cases, a star feed involves a visual change to one of the viewers 48. For example, a star feed will create/write registry values to reflect a change in advertisement on a particular viewer 48 (step 2). Star feeds are thus special feeds in that they can change register keys which point to bitmap files, source names, URL sources and so forth. In particular, referring to FIG. 23, star feeds are received by the communications server 38 and passed to the user interface alert panel 50 for processing. The registry values updated by star feeds are read by other components and the changes programmed by the star feeds are then put into effect. In operation, the user interface alert panel 50 first determines if a message is a star feed by checking the message tag to determine if it contains the star feed indicator, preferably "*=". It then parses the star feed extracting the component code and the registry key values to be updated. The updated key values are then written to the registry 49 where they are accessed by other components, such as the remote control 54

6,021,433

25

and the viewers 48. The basic structure of a star feed message is shown as follows:

<hr/>	
FEED_TAG~AV=COMPONENT_CODE~P=REGISTRY_KEY_VALUES where	
FEED_TAG=	the message tag code ("*" for star feeds)
COMPONENT_CODE=	a two letter code indicating to which component the star feed applies (e.g., BB for baseball viewer, RC for remote control, etc.)
REGISTRY_KEY_VALUES=	one or more sequences of the following parameters for the specified component: registry key, full file path name flag (0 or 1) if the key value contains a file name, and the registry key value.
<hr/>	

In a typical example, bitmaps for the Internet baseball score button are changed as well as the URL for the source:

<hr/>	
*~V=BB~P=Ad1;0;shared\bmps\SprtNet.bmp TV B;0;shared\bmps\SprtNetU.bmp ADB;0;shared\bmps\SprtNet.bmp ADB;0;shared\bmps\SprtNetU.bmp Ad1U;2;http://www.sportsnetwork.com:	
<hr/>	

In the example, new bitmap files SprtNetU.bmp, SprtNet.bmp and new URL http://www.sportsnetwork.com are added to the registry settings for the Baseball viewer. Where a new bitmap or other file name is specified in a star feed, the new file will have been previously received from the wireless broadcast network by the communications server 38 via the binary file transfer capability. This process is transparent to the user.

If in step 1, it is determined that the feed is a binary type feed, the binary feed is converted to a common EMIT string format (step 3). When the message is in the EMIT string format, a record is added to the message database by first determining the preferred viewer for the feed (step 4) and then by parsing out the EMIT string to common viewer fields (step 6).

In particular, to determine the preferred viewer for the feed (step 4), a filter field from the SYSAPPS table is compared to the EMIT string (step 5). In a typical configuration, approximately thirty viewers 48 are available and the user interface alert panel 50 determines which viewer 48 will be able to read the information. The preferred viewer is the actual icon which will fly up to the user interface alert panel 50. To obtain a viewer alias match, the user interface alert panel 50 obtains the necessary information by looking at the systems applications (SYSAPPS) table or database. By comparing a filter field from the SYSAPPS database to the EMIT string, the user interface alert panel 50 determines which viewer 48 is the preferred viewer and which viewer 48 should fly up to the user interface alert panel 50. For example, for a football related message, the filter fields from the SYSAPPS database would be reviewed against the football related message to determine the viewer alias match.

In accordance with the present invention, level tags further define the EMIT message so when the comparison is executed in SYSAPPS table, it can be determined which feed is for which viewer (level tag 1-5). A typical sample preferred filtering string is as follows:

1=N,2=N,N=*,R!=*,1=N,2=N,h=*,R!=*

26

Under the sample preferred filtering string, the level tags are 1=N, 2=N. By comparing 1=N, 2=N against the sample EMIT feed, it knows that this is a news marquee feed.

After a viewer alias match is achieved, a "Q" time flag or time flag reflecting the local time at which the message arrived at a user is created (step 6). The EMIT string is then parsed into common viewer fields and written to a message database 51 (step 8). The common fields include but are not limited to level tags, data and time, titles, source and content.

In the VGROUPS, there is a description for each viewer—a text typed out in a particular field. If you put the mouse over one of the buttons on the alert panel, on the bottom, it will say what this is. That description is pulled from VGROUPS (step 8).

After the EMIT feed is recorded to the message database 51 (step 8), the message is broadcast to the preferred viewer via the viewer server (steps 9-14). Initially, it is determined whether the viewer is running (step 9). If the viewer is running, e.g. football viewer is already running, the message is sent directly to the viewer server (step 10).

If the viewer is not running, it is determined whether the viewer should be auto launched (step 11). If auto-launch has been turned on for this viewer, then the viewer with message playback is launched. For example, for a football type feed, the viewer preferences are reviewed and if the user is setup for automatic launch of football, the football viewer with message playback is launched (step 12).

If the preferred viewer is not running, the fly-in sequence comprising a) creating a fly-in animation object, b) playing a viewer specific wave file, c) animating a button on the user interface alert panel 50, and d) placing a static button on the user interface alert panel 50, is initiated (step 13). In particular, a fly-in animation object is initially created. The fly-in animation object is an actual icon shown flying in from the opposite edge to the user interface alert panel 50. In accordance with an advantage of the present invention, fly-ins alert the user that new data is available for viewing. Fly-ins are small windows displaying animated graphics representing a particular message type, e.g. E-mail, which moves from the bottom right part of the user display screen to the user interface alert panel 50 whenever a new message of that particular type is received. If the user interface alert panel 50 is in a floating state, then the fly-in animation objects flies in from a random edge. At the same time the fly-in occurs, a viewer specific sound wave file is initiated. A button is then animated on the user interface alert panel 50. Finally, a static button which the user can press to launch the viewer is placed on the user interface alert panel 50 (step 13) and when depressed (step 14), will launch a viewer with message playback (step 12). For example, for a football feed, a fly-in animation object in the form of a football lands on top of the user interface alert panel 50, a trumpet will play followed by a button animated on the alert panel 50, which typically spins around and finally a static button appears on the alert panel 50. Fly-in graphic and default sound effects reflect message type. For example, for a golf feed, a golf tournament fly-in includes an image of a golf ball and the sound of a golf ball falling into a cup.

When the static button on the user interface alert panel is pressed (step 13), the viewer with message playback option is launched (step 12). The message is sent to the viewer server 20 which is the actual application which physically launches the viewer 48.

(iii) Timely events

The user interface alert panel will periodically and automatically perform the following functions: (1) check messages that require a mark for deletion, (2) check for valid

6,021,433

27

service plans, (3) check for delayed broadcasts, and (4) implement fly-in graphics for new messages, each of which is described in detail below.

(1) Check messages that require a mark for detection.

Each viewer has an entry in the SYSAPPS table that specifies the lifetime of the messages. A comparison is made to the message database and if a record needs to be marked for deletion, an "X" is placed in the MSG_READ field. In a preferred embodiment, this function is performed every 24 hours. The user interface alert panel 50 will decide, based on the information in the SYSAPPS table, how long a message should be kept for a particular viewer 48. For example, for a football viewer, if it is only desirable to see messages 2 days old, the user interface alert panel 50 will check against that field and when 2 days has transpired, proceed to mark those records for deletion.

(2) Check for valid service plans

The user interface alert panel 50 will also periodically check for valid service plans. Service plans typically dictate what kinds of feeds are available to a user. All valid plans are recorded in the registry so that other modules can read the information. The service plan checking preferably occurs at initialization and every 5 minutes thereafter. The user is also prompted with "plan expiration reminders."

(3) Check for delayed broadcasts

The user interface panel 50 also checks for delayed broadcasts which allow messages to be submitted for future broadcast. If a date and time has arrived for a delayed message, the MSG_READ field will be changed from "B" to "N" and a button will be placed on the user interface alert panel 50. Delayed broadcasts are preferably checked every five minutes. The user interface panel 50 thus checks every 5 minutes for special records that need to be shown to the user and then will change a particular field in the message database—the "B" to "N" so that next time it will not rebroadcast the same message again.

(4) Implement fly-in graphics means for new messages

The user interface alert panel 50 performs fly-in graphics for new messages received from the communications server 38 if this option has been selected by the user.

e. Viewer Server

Referring to FIG. 10, the viewer server 20 provides the means by which other components can initiate the execution of viewers 48 to display messages received from the broadcast network. This includes launching a particular viewer 48 upon command, parsing messages, and providing data extracted from the messages to the viewers 48 for display. The viewer server 20 also acts as the interface between the viewers 48 and the messages data base 51. Functionality of the viewer server 20 is accessed through the Viewer Server Applications Programming Interface (VSAPI).

The viewer server 20 serves the global control preferences across all viewers and allows common controls to be shared by viewers requiring similar functions. In accordance with the present invention, three different classes of user interface are present. One class, the viewer class, views a particular type of information, such as baseball or electronic mail. A second class, the viewer controller, is able to start and stop the other class, the viewers class. For example, in operation, the viewer controller resembles a remote control and enables a user to turn the viewers on and off. By utilizing the remote control, a user can thus automatically bring up a baseball viewer and baseball information will be automatically displayed in that viewer. For illustrative purposes, FIGS. 24(a), (b), (c) and (d) are depictions of a market scoreboard viewer, a football viewer, a newspaper viewer and stock ticker viewer, respectively.

28

In particular, in accordance with the feed format of the present invention, information is broken into logical information categories at the central broadcast server 34 and which matches viewers 48 which exist on the user end. The viewer server 20 ties into the viewers 48 so that an actual feed, such as an electronic mail notification feed, baseball sports feed or headline feed, is established. In accordance with the present invention, the data at the server end is classified into various formats to be able to indicate what type of a feed is present. This is accomplished by placing tags in front of various words that break it up into a type of information, such as a headline story, electronic mail story, financial story, and the like. This is the basis of the EMIT format which was described previously.

When this data arrives on the user side, the viewer server 20 reads the message including the codes and determines what type of message is being sent. Thus a viewer that is capable of displaying baseball information only receives baseball information.

In accordance with an alternative embodiment of the invention, another viewer controller which enables both incoming information as well as past information to be viewed can be utilized. Thus, for example, a user can bring up a baseball game that occurred earlier in the day. In operation, the viewer controller talks to the viewer server 20 and indicates that it wants to bring up a particular viewer. The viewer server 20 then activates and launches that particular viewer.

Preference viewers enable each of the viewers in a common user interface to show any preference information it has. The preferences viewers can be programmed to provide various kinds of information. For example, the preferences viewer can be directed to information relating to baseball teams. Another preferences viewer can be directed to stock market information. The preferences viewer can be further programmed to provide indication of events which are currently happening. For example, if the price of a stock, such as IBM, goes above a certain amount, such as \$100.00 per share, a stock market crawl viewer will come up to the foreground immediately and flash a red light.

f. Remote Control

The remote control 54, as shown in FIG. 11, provides a user interface for opening, closing and controlling viewers (viewer management), for maintenance of user settings and preferences, and for viewing the latest broadcast network news. It also maintains a message history log which allows the user to view previously received messages. Viewer control functions include mute, pause and volume level control for the viewer audio device. The remote control 54 is launched through the user interface alert panel 50.

g. Viewers

Viewers 18, opened through the user interface alert panel 50 or remote control 54, are the means by which data received from the broadcast network is displayed to the user. There are separate viewers for each of the different types of information provided over the network. Viewers 48 are capable of reading and displaying various message formats and contain preferences governing viewer actions. Viewers generally include, but are not limited to, graphics, data, sound files, and launch icons.

When each of the viewers 48 is installed, it goes through a registration process with the viewer server 20 and the viewer server 20 stores entries in the database that keep track of each of the viewers by way of the viewer table. A filtering means is provided for each viewer for filtering particular types of messages a viewer can look at. For example, a baseball viewer who wants to look at messages relating to

6,021,433

29

baseball information has two filtering means—one for saving information in a database and another filter for indicating that this is the type of information that should immediately be brought up to the viewer. Thus, if a viewer is interested in Dodger baseball games, such games would instantly be brought up by the second filter. Moreover, if a viewer desires to save all of the games in the national league, the filter for saving such information would be implemented

h. User Preferences Dynamic Link Library (DLL)

The User Preferences Dynamic Link Library (DLL) **53** allows the user to precisely specify what information is to be displayed by the Viewers **48** and how this information will be displayed and enters various related information, such as, the name of the user's Internet browser and activation codes for activating service plans. For example, the user can select the teams for which baseball or football scores will be shown, the sources of news stories, and the speed at which text is scrolled in Marquee type viewers. The User Preferences DLL **53** is accessed via the remote control **54** or through any open viewer **48**.

i. Address Reprogramming and Activation Code Parsing DLL

The address reprogramming and activation code parsing DLL **57** parses and validates service plan activation codes received over the wireless broadcast network or entered by the user and address reprogramming messages received over the network. Activation codes and address reprogramming messages control what broadcast network messages the user is allowed to receive. The code parsing DLL is used by the communications server **38**, remote control **54** and user preferences DLL **53**.

j. Error Logging

Error Logging **55** provides a means by which all other components can record the occurrence of errors or potential problem conditions in a log file. The error log can be a valuable aid to technical support in diagnosing problems a user may encounter in running software. The log file is preferably in ASCII text format and can be viewed by any word processor or text editor, such as, Microsoft Word or Notepad.

k. Operation of Received Message Data Flow

In operation, when a new message is received from the broadcast network, the communications server **38** receives a new data block from the wireless device **42** via the driver **44** and wireless interface **46**. Depending on the data block type, the communications server **38** either processes it locally or passes it to the user interface alert panel **50**. The user interface alert panel **50** receives a data block from the communications server **38**, stores it in the messages data base **51**, displays an icon for the particular message type and generates a fly-in or other means for notification such as an audio and/or visual alert for the new message if that option is selected by the user. If the user clicks on the icon for the new message, the user interface alert panel **50** sends a command to the viewer server **20** to open the appropriate viewer **48** to display the contents of the message. Alternatively, a viewer **48** to display the new message can be launched through the remote control **54**. Upon receiving the command to open a viewer **48**, the viewer server **20** parses the message, launches the viewer **48** and passes to it the data to be displayed. The viewer **48** displays the message data received from the viewer server **20** and commands the viewer server **20** to mark the message as "read" in the data base. At any step in the process, if an error condition is detected, it is recorded in the error log **55**.

1. E-mail Alerts

FIG. **13** is a flow chart of an algorithm for generating and processing E-mail alerts in accordance with the present

30

invention. In accordance with the present invention, a user may be instantly notified of E-mail messages without being connected to an E-mail service provider. Referring to FIG. **13**, when a user receives an E-mail message (step **240**), the user's provider sends an E-mail notification to central broadcast server (step **244**). Upon receiving this notification, the central broadcast server transmits an E-mail alert message to the user's computer through the broadcast network (step **246**). When the alert message is received by the software application in the user's computer, an animated visual and/or audio notification is triggered, or the e-mail viewer automatically pops up, depending on the mode of operation selected by the user (step **248**). In the first case, an E-mail alert icon appears on the alert panel and the E-mail viewer can be launched in the same manner as viewers for news alerts (i.e. by clicking the icon or through the remote control). An E-mail alert contains the provider ID code number and the "From" name (E-mail address of the sender). One skilled in the art will recognize that the alert is not limited to the provider ID code number and name. Rather, the E-mail alert could include a header, whole message etc. The E-mail viewer displays an icon corresponding to the provider ID, the date and time the alert was received, and the sender's E-mail address. To read an E-mail message, the user simply clicks the associated icon (step **250**) which causes the E-mail program for the particular provider to be launched (step **252**). The user's E-mail can then be retrieved through a wired connection to the E-mail provider (step **254**). One skilled in the art will recognize that E-mail alerts may be received from more than one source. For example, a user may receive an E-mail alert from an Internet E-mail provider and America On-Line or CompuServe.

User Wireless On-Line Guide

In accordance with the present invention, a wirelessly transmitted on-line guide provides a detailed schedule of when certain information, such as upcoming events, forums and chat sessions, will be transmitted. With ongoing wireless broadcasts, the information in the on-line guide is maintained up-to-date. In particular, the on-line guide can notify a user just before an event is about to happen on the Internet, therefore eliminating the need to manually keep track of upcoming events. The user indicates which events are important, and the on-line guide reminds the user via an alarm including a visual and sound alert of the events at a predetermined time, such as minutes, before each occurs. The user can then click on the event and a connection to the event's location on the Internet is made through the user's standard Internet browser and Internet service provider. Alternatively, a user can specify that a connection to the event location via the user's Internet browser and Internet service provider be made automatically when the selected event is about to occur.

URL Broadcast and Hot Links

Referring to FIG. **1**, the URL broadcast and hot links **22** back to the information source **12** is shown. In accordance with the present invention, very short notification centric messages such as news headlines from information sources **12**, such as Internet, on-line services and other information providers, are transmitted to the computer **14** by wireless transmission. A user, from a computer **14**, can make a wired connection **24** back to the information source **12** to obtain more detailed information. In accordance with the present invention, attached to each of the notification centric messages is a universal resource locator (URL) code **22** as well as related Internet address information. This allows the user, by clicking on an icon that is embedded in the message, to

6,021,433

31

make a wired or wireless connection **24**, either through a modem, TC/IP or LAN-type connection, and automatically establish a link back to the information source **12**. The user can thus go directly to the specific site that the information came from. In a typical example, the specific site can be ten pages deep. Thus, in accordance with an advantage of the present invention, information sources **12** such as the Internet and other on-line services, which are typically overwhelming particularly with respect to locating a story, are easily accessible. The present invention allows a user to pinpoint and locate the specific information the user was alerted to. The user can thus hit one button which establishes the connection **24** and takes the user directly to the location where the information is located.

FIG. **12** is a flow chart of an algorithm for extracting and processing the Internet source URL for messages broadcast over the wireless communication network illustrated in FIG. **1**. In accordance with the present invention, the Internet source for a news item alert is broadcast as part of the alert message itself (step **260**). The message contains a number of tags delineating the various parts of the message. In the preferred embodiment, tags "S=" and "U=" identify the Internet source where detailed information about the news alert may be found. For those messages which always originate from the same list of default sources, the "S" tag only applies (step **264**). Following the "S=" tag is a letter code corresponding to the Internet URL. For example, the letter code for an alert from the Reuters News Service is "R". The actual URL, <http://www.reuters.com>, is obtained by using the letter code as an index into the alert source database of the present invention (step **266**). URL's in the alert source database may be updated by Star Feed messages in case changes in the default URL's are necessary (step **268**). For messages whose sources are not limited to a default set, the "U" tag conveys the Internet source (step **272**). Following the "U=" tag is the actual URL source of the message (e.g. U=<http://www.universalnews.com>). Wireless throughput is conserved by transmitting the full URL only in those cases where the source is not restricted to being a member of a fixed set. The source URL is displayed at the end of the alert message text (step **270**). A user with a wired or wireless connection to the Internet can go directly to the alert source simply by clicking the URL (step **270**). A connection to the alert source on the Internet is thus provided.

Over the Air Programming

Services received and various operational characteristics at the user end can be programmed by the central broadcast server **34** through the wireless broadcast network. This is accomplished primarily through Star Feeds and service activation/deactivation codes. Star Feeds, which have been described in detail above, are special messages which allow parameters controlling viewer operation to be modified from the central broadcast server **34**. Activation/deactivation codes determine which services a user is allowed to receive. For example, if a user subscribes to e-mail alerts, this service can be turned on for that specific user through an e-mail alert activation code message transmitted to the user site via the wireless broadcast network. Conversely, if a user stops subscribing to a service, that service can be turned off through a deactivation code message. Additionally, the capability exists for binary file transfer from the central broadcast server **34** to add new executable files or replace existing ones with newer versions. In this way, new or updated viewers can be installed directly through the wireless broadcast network.

32

Billing and Activation Server

Referring to FIG. **1**, users may remotely request additional services or modify existing services from the personal computer **14** or other computing device through a billing and activation server **64** which communicates with the central broadcast server **34**. By telephone or modem communication, a user can contact the billing and activation server **64** which in turn communicates with the central broadcast server **34**. Once such a request has been processed by the central broadcast server **34**, the server **34** wirelessly transmits an activation code directly to the message server **18** to activate additional or modify existing services. By matching the serial number contained in the broadcast message with the users serial number, the user software will program a receiver board in the user receiver **32** to begin receiving additional or modified services. Thus according to an advantage of the present invention, users can remotely adjust services from their personal computers **14** or other computing devices.

Simultaneous Wired Transmission

In accordance with an alternate embodiment of the invention, the information provided from the information sources **12** and transmitted to the central broadcast server **34** to be consolidated in accordance with the present invention and then transmitted wirelessly nationwide to personal computers **14** and other computing devices as described in detail above can also be sent simultaneously via a wired connection to the same personal computers **14** and computing devices having Internet/World Wide Web access (direct or via on-line service providing Internet and World Wide Web access). In particular, the data processed at the central broadcast server **34**, in addition to being transmitted wirelessly, is simultaneously placed on Web pages on the Internet. A user can thus connect to the Web via the Internet. In operation, to access data sent by the central broadcast server **34**, a user makes a connection via the Internet to the World Wide Web server and delivers its URL request. The request is acknowledged by the Web server, which then sends the requested data to the user. Thus, a user can receive real time data/information in the form of voice, video data or a combination thereof by accessing the World Wide Web.

It will be appreciated by persons skilled in the art that the present invention is not limited to what has been shown and described hereinabove, nor the dimensions of sizes of the physical implementation described immediately above.

What is claimed is:

1. A method for transmitting data to selected remote computing devices, comprising the steps of:
 - transmitting data from an information source to a central broadcast server;
 - preprocessing said data at said central broadcast server, further comprising the step of:
 - providing data to servers in said central broadcast server;
 - parsing said data with parsers corresponding to said servers;
 - transmitting said data to a content manager for determining how said data is handled;
 - transmitting said data from said content manager to an information gateway for building data blocks and assigning addresses to said data block; and
 - transmitting said data blocks from said information gateway to a transmission gateway for preparing said data block for transmission to receivers;
 - transmitting preprocessed data to receivers communicating with said computing devices; and
 - instantaneously notifying said computing devices of receipt of said preprocessed data whether said computing devices are on or off.

6,021,433

33

2. The method claimed in claim 1, wherein said step of transmitting said data from said content manager to said information gateway for building data blocks and assigning addresses to said data block, further comprises the step of:

building data blocks and assigning addresses to said data block based on information in a subscriber database.

3. The method claimed in claim 1, wherein said step of transmitting preprocessed data to remote receivers communicating with said computing devices, further comprises the step of:

wirelessly transmitting said preprocessed data to remote receivers.

4. The method claimed in claim 3, wherein said step of wirelessly transmitting said preprocessed data to remote receivers further comprises the step of:

transmitting said preprocessed data utilizing a paging network.

5. The method claimed in claim 3, wherein said step of wirelessly transmitting said preprocessed data to remote receivers further comprises the step of:

transmitting said preprocessed data utilizing a Vertical Blanking Interval.

6. The method claimed in claim 3, wherein said step of wirelessly transmitting said preprocessed data to remote receivers further comprises the step of:

transmitting said preprocessed data utilizing a satellite system.

7. The method claimed in claim 3, wherein said step of wirelessly transmitting said preprocessed data to remote receivers further comprises the step of:

transmitting said preprocessed data utilizing a FM subcarrier, digital, analog, cellular, GSM or PCS carrier.

8. The method claimed in claim 1, wherein said step of transmitting preprocessed data to remote receivers communicating with said computing devices, further comprises the step of:

transmitting said preprocessed data to remote receivers by wired transmission.

9. The method claimed in claim 1, wherein said step of preprocessing data at said central broadcast server, further comprises the step of:

attaching to said preprocessed data an Internet address location of said preprocessed data for providing to said user an automatic connection back to said information source for obtaining further information related to said preprocessed data.

10. The method claimed in claim 9, wherein said Internet address location is a Uniform Resource Locator.

11. The method claimed in claim 9, wherein said step of attaching to said preprocessed data an Internet address location of said preprocessed data for providing to said user an automatic connection back to said information source for obtaining further information related to said preprocessed data, further comprises the step of:

providing an automatic connection back to said information source through an user activating a single function on said computing device.

12. The method claimed in claim 11, wherein said single function comprises a single click on said computing device.

13. The method claimed in claim 9, wherein said connection back to said information source for obtaining further information related to said preprocessed data is an automated wired connection.

14. The method claimed in claim 9, wherein said connection back to said information source for obtaining further

34

information related to said preprocessed data is an automated wireless connection.

15. The method claimed in claim 9, wherein said step of attaching to said preprocessed data an Internet address location of said preprocessed data for providing to said user an automatic connection back to said information source for obtaining further information related to said preprocessed data, further comprises the step of:

determining at said central broadcast server said Internet address location from said information source.

16. The method claimed in claim 9, wherein said step of attaching to said preprocessed data an Internet address location of said preprocessed data for providing to said user an automatic connection back to said information source for obtaining further information related to said preprocessed data, further comprises the step of:

attaching said Internet address location to said preprocessed data.

17. The method claimed in claim 9, wherein said step of attaching to said preprocessed data an Internet address location of said preprocessed data for providing to said user an automatic connection back to said information source for obtaining further information related to said preprocessed data, further comprises the step of:

transmitting said Internet address location with said preprocessed data to said computing device.

18. The method claimed in claim 9, further comprising the step of:

extracting said Internet address location from said preprocessed data at said computing device.

19. The method claimed in claim 9, further comprising the step of:

displaying said Internet address location with said preprocessed data to said user such that said user can with a single click on said Internet address location to obtain additional information from said information source.

20. The method claimed in claim 9, further comprising the step of:

launching an Internet browser and passing said Internet address location to said browser for automatic connection back to said information source.

21. The method claimed in claim 9, wherein said Internet address is a proprietary on-line addressing scheme.

22. The method claimed in claim 1, wherein said step of instantaneously notifying said computing devices of receipt of said preprocessed data whether said computing devices are on or off, further comprises the step of:

providing at least one alert which when activated allows display of data.

23. The method claimed in claim 22, wherein said at least one alert comprises a visual alert.

24. The method claimed in claim 22, wherein said at least one alert comprises an audio alert.

25. The method claimed in claim 22, wherein said at least one alert is related to type of information present at computing device.

26. The method claimed in claim 1, wherein said step of instantaneously notifying said computing devices of receipt of said preprocessed data whether said computing devices are on or off, further comprises the step of:

providing a dockable user interface alert panel on a display communicating with computing device for providing alerts to said user, wherein said alert panel is dockable on top of other applications.

27. The method claimed in claim 26, wherein said step of providing a dockable user interface alert panel on a display

6,021,433

35

communicating with computing device for providing alerts to said user, further comprises the step of:

displaying fly-in graphics and icon buttons to alert said user that new data has been received by said computing device.

28. The method claimed in claim 1, wherein said step of preprocessing said data at said central broadcast server further comprises the step of:

deriving redundant data packets for transmission to said user.

29. The method claimed in claim 28, wherein said step of deriving redundant data packets for transmission to said user further comprises the step of:

parceling a data block into at least one incoming message.

30. The method claimed in claim 29, wherein said step of deriving redundant data packets for transmission to said user further comprises the step of:

parceling said messages into k information packets.

31. The method claimed in claim 30, wherein said step of deriving redundant data packets for transmission to said user further comprises the step of:

selecting a number of parity-check packets p.

32. The method claimed in claim 31, wherein said step of deriving redundant data packets for transmission to said user further comprises the step of:

encoding column-wise with a modified Reed-Solomon code for generating parity-check packets.

33. The method claimed in claim 32, wherein said Reed-Solomon code is defined in accordance with:

$$g(x) = \prod_{i=1}^p (x + a^i).$$

34. The method claimed in claim 32, wherein said step of deriving redundant data packets for transmission to said user further comprises the step of:

parceling said data packets into code words for transmission to said user.

35. The method claimed in claim 34, wherein said step of deriving redundant data packets for transmission to said user further comprises the steps of:

performing error correction and detection on said code words after said data packets have been parceled.

36. The method claimed in claim 34, further comprising the step of:

assembling a data block from said code words.

37. The method claimed in claim 36, wherein said step of assembling a data block from said code words further comprises the step of:

counting the number of code words which have errors;
determining whether each packet has any errors;
saving packets without error;
discarding packets with at least one error; and
assembling a message when the required number of packets has been received.

38. The method claimed in claim 28, wherein said data packets include information packets and parity-check packets.

39. The method claimed in claim 1, wherein said step of preprocessing said data at said central broadcast server further comprises the step of:

combining Huffman compression and the dictionary-based compression based algorithms.

36

40. The method claimed in claim 39, wherein said step of combining Huffman compression and the dictionary-based compression based algorithms further comprises the steps of:

scanning input texts;

searching for next item previously seen text;

searching for next item in a static Huffman dictionary; and
choosing said search method which produces a better result for compression.

41. The method claimed in claim 40, further comprising the step of:

decompressing said compressed data.

42. The method claimed in claim 1, wherein said step of preprocessing said data at said central broadcast server further comprises the step of:

utilizing a differencing algorithm for compressing said coded data, thereby significantly reducing the number of bytes sent with each transmission.

43. The method claimed in claim 1, wherein said step of preprocessing data at said central broadcast server, further comprises the step of:

processing data in accordance with feed type from said information source.

44. The method claimed in claim 43, wherein said feed type comprises binary type feeds.

45. The method claimed in claim 43, wherein said feed type comprises common user information type feeds.

46. The method claimed in claim 43, wherein said feed type comprises feeds for modifying registry keys which control processing of data.

47. The method claimed in claim 43, wherein said step of processing data in accordance with feed type from said information source, further comprises the step of:

using tags to differentiate types of information.

48. The method claimed in claim 1, wherein said step of instantaneously notifying said computing devices of receipt of said preprocessed data whether said computing devices are on or off, further comprises the step of:

instantaneously alerting said user to personal alerts through the use of sound, graphics, bit maps or video, wherein said user can instantaneously access information.

49. The method claimed in claim 1, wherein said step of preprocessing data at said central broadcast server, further comprises the step of:

encoding said data with information relating to message parameters for filtering.

50. The method claimed in claim 1, wherein said step of instantaneously notifying said computing devices of receipt of said preprocessed data whether said computing devices are on or off, further comprises the steps of:

monitoring said transmissions utilizing multiple viewers;
filtering said transmitted preprocessed data;
post processing said preprocessed data; and
notifying said user instantaneously of receipt of filtered postprocessed data.

51. The method claimed in claim 50, wherein said step of filtering said transmitted preprocessed data further comprises the step of:

filtering said transmitted preprocessed data in accordance with preferences set by said user.

52. The method claimed in claim 51, wherein said step of filtering said transmitted preprocessed data in accordance with preferences set by said user, further comprises the step of:

6,021,433

37

setting said preferences with respect to sound, video and animation.

53. The method claimed in claim 50, wherein said step of filtering said transmitted preprocessed data further comprises the step of:

filtering said preprocessed data in accordance with virtual addresses.

54. The method claimed in claim 50, wherein said step of filtering said transmitted preprocessed data further comprises the step of:

filtering said preprocessed data in accordance with physical addresses.

55. The method claimed in claim 50, further comprising the step of:

controlling said viewers from said central broadcast server.

56. The method claimed in claim 55, wherein said step of controlling said viewers from said central broadcast server, further comprises the step of:

adding viewers from said central broadcast server.

57. The method claimed in claim 55, wherein said step of controlling said viewers from said central broadcast server, further comprises the step of:

removing viewers from said central broadcast server.

58. The method claimed in claim 50, further comprising the step of:

utilizing a remote control interface for controlling said viewers.

59. The method claimed in claim 58, wherein said step of utilizing a remote control interface for controlling said viewers further comprises the step of:

launching said remote control interface through a user interface alert panel.

60. The method claimed in claim 50, further comprising the step of:

storing entries in a viewer server connected to said viewer; and

providing filtering means for filtering particular types of messages a viewer can look at.

61. The method claimed in claim 1, further comprising the step of:

activating said preprocessed data at a scheduled time.

62. The method claimed in claim 1, further comprising the step of:

modifying said preprocessed data instantaneously and wirelessly.

63. The method claimed in claim 56, wherein said step of modifying said preprocessed data instantaneously and wirelessly, further comprises the step of:

activating services wirelessly through activation codes which enable or disable services.

64. The method claimed in claim 1, further comprising the step of:

postprocessing said preprocessed data.

65. The method claimed in claim 64, wherein said step of postprocessing said preprocessed data further comprises the step of:

recombining, decoding and decompressing said preprocessed data.

66. The method claimed in claim 1, wherein said information source may be an Internet access provider providing data feeds.

67. The method claimed in claim 1, wherein said information source may be an on-line service provider providing data feeds.

68. The method claimed in claim 1, wherein said step of transmitting said data from said content manager to said information gateway for building data blocks and assigning addresses to said data block, further comprises the step of:

38

building data blocks and assigning addresses to said data block based on information in a subscriber database.

69. The method claimed in claim 1, further comprising the step of:

displaying contextual graphics on said computing device to show data in a predefined format.

70. The method claimed in claim 69, wherein said predefined format is a scoreboard.

71. The method claimed in claim 1, wherein said step of preprocessing data at said central broadcast server, further comprises the step of:

attaching to said preprocessed data an Internet address location of said preprocessed data for providing to said user a message that causes a process or transaction on said computing device to occur.

72. The method claimed in claim 1, wherein said step of preprocessing said data at said central broadcast server, further comprises the step of:

sending said data on groups of pooled capcodes.

73. The method claimed in claim 72, wherein said step of sending said data on groups of pooled capcodes, further comprises the step of:

multiplexing data over multiple capcodes to be reassembled at said user as if data were being sent over a single capcode.

74. The method claimed in claim 1, wherein said step of preprocessing said data at said central broadcast server, further comprises the step of:

assigning data packets to a group of capcodes;

transmitting said data over a paging network using said group of capcodes;

receiving packets at said user on said group of capcodes; and

combining said packets from group of capcodes into one data message.

75. A method for transmitting data to selected remote computing devices, comprising the steps of:

transmitting data from an information source to a central broadcast server;

preprocessing said data at said central broadcast server;

transmitting preprocessed data to remote receivers communicating with said computing devices;

instantaneously notifying said computing devices of receipt of said preprocessed data whether said computing devices are on or off, further comprising the steps of:

monitoring said transmissions utilizing multiple viewers;

filtering said transmitted preprocessed data;

post processing said preprocessed data; and

notifying said user instantaneously of receipt of filtered postprocessed data; utilizing a remote control interface for controlling said viewers.

76. The method claimed in claim 75, wherein said step of utilizing a remote control interface for controlling said viewers further comprises the step of:

launching said remote control interface through a user interface alert panel.

77. The method claimed in claim 76, further comprising the step of:

storing entries in a viewer server connected to said viewer; and

providing filtering means for filtering particular types of messages a viewer can look at.

* * * * *